

GRAVITATIONAL WAVE CLASSIFICATION USING DEEP LEARNING

**Thesis submitted in fulfilment of the requirements
for the degree of
Bachelor of Science
in Pure Physics**

by

NG WEI YI



School of Physics

UNIVERSITI SAINS MALAYSIA

JULY 2022

PREFACE

I, Ng Wei Yi, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

All the work described in this thesis was carried out in the School of Physics in Universiti Sains Malaysia, under the supervision of Dr. John Soo Yue Han. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others except as specified in the text and summarized in the Acknowledgment.

The thesis is not substantially the same as any that I have submitted or is being concurrently submitted, for a degree or diploma or other qualification at Universiti Sains Malaysia or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed 100,000 words, including footnotes, tables, and figures but excluding bibliography, appendices, and supporting data.

ACKNOWLEDGEMENT

The process of writing this dissertation would not have been possible without the support of many people around me. I would like to start with by thanking my supervisor, Dr. John Soo Yue Han, whose expertise and patience carried me through all the stages of writing this dissertation and helped in clearing some of my confusions. I could not have imagined having a better advisor for this dissertation.

I would like to express my gratitude to Assoc. Prof. Dr. Yoon Tiem Leong for helping me in setting up one of the computers at School of Physics for me due to the limited computational resources of my machine to carry out this dissertation by using different deep learning models chosen in this project. Moreover, I would like to extend my gratitude to my project partner, Sharon Bin Malodol @ Jamilon of Universiti Sains Malaysia for the discussions we had with our project throughout this time which were quite helpful in progressing the dissertation. I would like to thank and appreciate Kaggle and G2Net for providing me the dataset of simulated gravitational waves to be used in this work.

In addition, I would want to express my gratitude to my parents and friends for their support in helping me getting through this dissertation as this is the first dissertation I have ever done. Although the obstacles were challenging, I believed in myself and faced those challenges without giving up.

TABLE OF CONTENTS

	Page
Acknowledgement.....	ii
Table of Contents	iii
List of Tables	vii
List of Figures	viii
Abstract	x
Abstrak	xi
 CHAPTER 1 – INTRODUCTION	
1.1 BACKGROUND	1
1.2 SCOPE OF STUDY	3
1.3 OBJECTIVES	3
1.4 THESIS LAYOUT	4
 CHAPTER 2 – THEORY AND LITERATURE REVIEW	
2.1 COSMOLOGY	6
2.1.1 Introduction	6
2.2 THE PRINCIPLES OF SPECIAL RELATIVITY	7
2.2.1 Galilean Transformations	7
2.2.2 Einstein’s Postulates and The Lorentz Transformations	9
2.3 CURVED SPACETIME OF GENERAL RELATIVITY	10
2.3.1 The Curvature of Spacetime	10
2.3.2 Principle of Equivalence and The Bending of Light	11

2.3.3	Spacetime Intervals	12
2.3.4	Einstein Field Equation	13
2.3.5	The Metric for Flat Spacetime.....	15
2.3.6	Curved Spacetime and Schwarzschild Metric	17
2.4	BLACK HOLES.....	18
2.4.1	The Schwarzschild Radius.....	19
2.4.2	Journey into a Black Hole	20
2.5	GRAVITATIONAL WAVES	21
2.5.1	A Linearized Gravitational Wave.....	22
2.5.2	Detecting Gravitational Waves	26
2.5.3	Gravitational Wave Polarization.....	29
2.5.4	Power and Energy in Gravitational Waves.....	31
2.5.5	Gravitational versus Electromagnetic Waves	32
2.5.6	Gravitational Wave Interferometers	34
2.6	DEEP LEARNING	35
2.6.1	Convolutional Neural Network(CNN)	36
2.7	LITERATURE REVIEW	43
 CHAPTER 3 – METHODOLOGY		
3.1	DATA DESCRIPTION	47
3.2	CONSTANT-Q TRANSFORM.....	48
3.3	TRANSFER LEARNING.....	53
3.4	EFFICIENTNET	54
3.4.1	Architecture	54
3.4.2	Modified Architecture	59

3.4.3	Fine-tuning	61
3.5	RESNET	61
3.5.1	Architecture	61
3.5.2	Modified Architecture	64
3.5.3	Fine-tuning	66
3.6	DENSENET	66
3.6.1	Architecture	66
3.6.2	Modified Architecture	70
3.6.3	Fine-tuning	70
3.7	LOSS FUNCTION	71
3.8	PERFORMANCE EVALUATION METRICS.....	71
3.8.1	Confusion Matrix	72
3.8.2	Accuracy.....	72

CHAPTER 4 – RESULTS AND DISCUSSION

4.1	EFFICIENTNET-B0	74
4.1.1	Confusion Matrix	74
4.1.2	Accuracy.....	75
4.1.3	Loss Function	77
4.2	RESNET-101	78
4.2.1	Confusion Matrix	78
4.2.2	Accuracy.....	79
4.2.3	Loss Function	80
4.3	DENSENET-121	81
4.3.1	Confusion Matrix	81

4.3.2	Accuracy.....	82
4.3.3	Loss Function	83
4.4	DISCUSSION	84
 CHAPTER 5 – CONCLUSION AND FUTURE WORK		
5.1	CONCLUSION.....	86
5.2	FUTURE WORK.....	87
References.....		89

LIST OF TABLES

Table 2.1	Differences between gravitational waves and electromagnetic waves [1].	33
Table 3.1	Parameters of CQT1992V2 function in this work[2]	52
Table 3.2	EfficientNet-B0 baseline network [3].	58
Table 3.3	EfficientNet Performance Results on ImageNet [3].	59
Table 3.4	Modified architecture of EfficientNet-B0. The highlighted row is the original architecture without an FC layer.	60
Table 3.5	Architectures of different ResNet models [4].	63
Table 3.6	Error rates of single-model results on the ImageNet of ResNet models [4].	64
Table 3.7	Modified architecture of ResNet-101. The highlighted row is the original architecture without an FC layer.	65
Table 3.8	Architectures of different DenseNet models [5].	69
Table 3.9	Error rates on the ImageNet validation set with single-crop testing of DenseNet models [5].	69
Table 3.10	Modified architecture of DenseNet-121. The highlighted row is the original architecture without an FC layer.	70

LIST OF FIGURES

Figure 2.1	Inertial reference frames S and S' [6]	8
Figure 2.2	Distance separation of two test masses perturbed by a gravitational wave [1]	28
Figure 2.3	Effect of purely plus-polarized and purely cross-polarized gravitational waves on a circular array of test masses [7]	31
Figure 2.4	Setup of Michelson–Morley Experiment [8]	34
Figure 2.5	LIGO gravitational wave detector at Hanford, Washington [9].	35
Figure 2.6	Architecture of CNN [10].	37
Figure 2.7	Graph of ReLu activation function [11].	39
Figure 2.8	Graph of sigmoid activation function [12].	40
Figure 2.9	Sensitivity curves illustrating the ability of the neural network to identify BNS and BBH GW signals. The true alarm probability (sensitivity) is plotted as a function of the optimal signal-to-noise ratio (SNR) for false alarm probabilities 10^{-1} , 10^{-2} , and 10^{-3} [13].	44
Figure 2.10	Deep learning forecasting for binary neutron stars in advanced LIGO data. An astrophysically motivated sample of binary systems and signal-to-noise ratios show that deep learning identifies signals in real data up to 30 s before merger [14].	45
Figure 3.1	Histogram of the original and extracted training set file.	48
Figure 3.2	Time-frequency resolution of STFT and CQT [15].	49
Figure 3.3	Spectrograms of STFT and CQT [15].	50
Figure 3.4	Representative data sample in the G2Net dataset and the spectrograms obtained by applying the constant-Q transform.	53
Figure 3.5	Scaling network width for different baseline networks [3].	56

Figure 3.6	Residual learning: a building block [4].	62
Figure 3.7	A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input [5].	67
Figure 3.8	A deep DenseNet with three dense blocks [5].	68
Figure 3.9	Confusion matrix for binary classification [16].	72
Figure 4.1	Confusion matrix of EfficientNet-B0. The top left and right of the confusion matrix show TN and FP respectively. Whereas the bottom left and right show FN and TP respectively.	75
Figure 4.2	Training and validation accuracy of EfficientNet-B0.	76
Figure 4.3	Training and validation losses of EfficientNet-B0.	77
Figure 4.4	Confusion matrix of ResNet-101.	78
Figure 4.5	Training and validation accuracy of ResNet-101.	79
Figure 4.6	Training and validation losses of ResNet-101.	80
Figure 4.7	Confusion matrix of DenseNet-121.	81
Figure 4.8	Training and validation accuracy of DenseNet-121.	82
Figure 4.9	Training and validation losses of DenseNet-121.	83

GRAVITATIONAL WAVE CLASSIFICATION USING DEEP LEARNING

ABSTRACT

Ever since Albert Einstein (1879 - 1955) published his paper about the general theory of relativity, more and more of this theory has been proven by physicists with the help of major advancements in modern technology. Deep learning has also gained quite a reputation over the last 10 years due to its cost effectiveness and the ability to extract information from large amount of data. In this work, we demonstrated how some deep learning models such as EfficientNet-B0, ResNet-101, and DenseNet-121 can be used to classify gravitational wave signals from noises which are imperceptible to the naked eyes. By learning about various deep learning models, we can understand how they truly operate. Furthermore, by modifying and fine-tuning the models such as using new fully connected layers and applying transfer learning, we can achieve better performance and efficiency in performing such task. Final results show that we can achieve 78.90% accuracy for EfficientNet-B0, 78.10% accuracy for ResNet-101, and 77.90% accuracy for DenseNet-121.

PENGELASAN GELOMBANG GRAVITI MENGGUNAKAN PEMBELAJARAN DALAM

ABSTRAK

Sejak Albert Einstein (1879 - 1955) menerbitkan kertas kerjanya tentang teori umum relativiti, semakin banyak teori ini telah dibuktikan oleh ahli-ahli fizik dengan bantuan kemajuan besar dalam teknologi moden. Pembelajaran mendalam juga telah mendapat reputasi yang agak baik sejak 10 tahun yang lalu kerana keberkesanan kosnya dan keupayaan untuk mengekstrak maklumat daripada sejumlah besar data. Dalam kerja ini, kami menunjukkan bagaimana beberapa model pembelajaran mendalam seperti EfficientNet-B0, ResNet-101, dan DenseNet-121 boleh digunakan untuk mengklasifikasikan isyarat gelombang graviti daripada bunyi yang tidak dapat dilihat oleh mata kasar. Dengan mempelajari tentang pelbagai model pembelajaran mendalam, kita boleh memahami cara ia benar-benar beroperasi. Tambahan pula, dengan mengubah suai dan memperhalusi model seperti menggunakan lapisan bersambung sepenuhnya baharu dan menerapkan pembelajaran pemindahan, kami boleh mencapai prestasi dan kecekapan yang lebih baik dalam melaksanakan tugas tersebut. Keputusan akhir menunjukkan bahawa kami boleh mencapai ketepatan 78.90% untuk EfficientNet-B0, 78.10% ketepatan untuk ResNet-101 dan 77.90% ketepatan untuk DenseNet-121.

CHAPTER 1

INTRODUCTION

The passing of one of the greatest physicists, Albert Einstein on 18 April 1955 [17] was a loss to the world of physics. However, his legacies live on and still continue to be proven by physicists of our era till this day.

One of them is gravitational wave which has just been proven in the recent 10 years thanks to the sophisticated and sensitive scientific instruments. The discovery of gravitational waves excited physicists and won Barry C. Barish, Rainer Weiss, and Kip S. Thorne a Nobel Prize in Physics in 2017 [18].

1.1 BACKGROUND

Gravitational waves are disturbances in the curvature of spacetime, produced by masses in motion, which propagate away as waves at the speed of light [19] [1]. Sources of gravitational waves are usually two neutron stars or black holes orbiting each other, a spinning neutron star that has bumps on or imperfections in its spherical shape or when a supernova event occurs. Gravitational waves were first proposed by a mathematician named Henri Poincaré in 1905 and predicted by Albert Einstein in 1916 on the basis of general theory of relativity [18] [20]. Similar to electromagnetic radiation, gravitational waves can transport energy as gravitational radiation [21] [22].

The curvature of spacetime is distorted when gravitational waves pass through.

Spacetime is being stretched back and forth, the distance between two objects in that situation increases and decreases rhythmically at a frequency similar to the waves that pass through [1]. However, the magnitude of this effect is inversely proportional to the distance from the source. The further the source that produces gravitational waves, the smaller the distortion of spacetime curvature and the lower the frequency [23].

Gravitational waves were first observed from signal of the merging of two black holes with masses of 29 and 36 solar masses about 1.3 billion light-years away on 14 September 2015 [24]. The observation was then announced by the LIGO-Virgo collaborations on 11 February 2016 [18]. The mass of the merged black hole was 62 solar masses and it released more than 50 times the power of all the stars in the observable universe combined during the final fraction of a second of the merger. The gravitational waves emitted were equivalent to three solar masses [25] [26]. The signal was detected by both LIGO detectors in Livingston and Hanford and due to the angle between the two detectors and the source, there was a time difference of 7 milliseconds detected by both detectors. The source of the signal was from the Southern Celestial Hemisphere with a rough direction of the Magellanic Clouds [27].

Currently, both Virgo and LIGO mainly use a technique called matched filtering technique to detect gravitational wave signals. This technique builds a theoretical waveform template bank to match the monitored data and collects trigger signals as candidates for further verification. It is done by sliding the data of one detector in time by an amount which is larger than the typical gravitational wave travel time between detectors. A measure of coincident events after the time slides is used to estimate the search background and the significance of detection candidates [28] [29].

However, the computational cost of such technique in low latency will grow dramatically as the low frequency sensitivity of gravitational-wave detectors improves. This in turns, limits the data processing speed, increase data processing time, and it will have difficulty to meet the needs of real-time observation if a big template bank is used [29][30]. Hence, such technique can be replaced with deep-learning-based methods.

1.2 SCOPE OF STUDY

In this thesis, we will build convolutional neural networks (CNNs) that can classify an event as a GW (gravitational wave) or an ND (non-detection) from time-series data containing simulated gravitational wave signals obtained from *G2Net* in a Kaggle competition [31]. The dataset will be divided into training, test and validation sets with a numerical response variable with a value of 1 for GWs and 0 for NDs.

1.3 OBJECTIVES

This study consists of several objectives as listed below:

1. To classify gravitational wave signals from background noises in a simulated time-series data (Obtained from Kaggle.com).
2. Compare three convolutional neural network (CNN) models and choose the champion model to detect gravitational wave signals from a time-series data that contain noises.
3. To provide an in-depth study of the issues that machine learning may be expected

to solve in the context of a search for gravitational waves, as well as to discuss their limitations in replacing matched filtering technique.

4. To demonstrate the CNN model's robustness for the time-series data with different architectures.

1.4 THESIS LAYOUT

This thesis is mainly about using different deep learning models to classify gravitational wave signals from background noises then choose the champion model for future use. The applications of deep learning have been increasing for the past 10 years, it is time we put it to good use.

This thesis is written as follows. In Chapter 1, we are briefly introduced the background of gravitational waves, when they were predicted and first detected, as well as why deep learning should be introduced to replace the old technique used to detect gravitational wave signals. In the latter part of Chapter 1, we discuss the objectives of this thesis and its scope of study.

In Chapter 2, we discuss the theories required for this thesis, which are special theory of relativity, general theory of relativity, black hole, gravitational waves, and deep learning. We will discuss how special and general relativity are used to build the foundation for the prediction of gravitational waves. Furthermore, deep learning is briefly being discussed to understand how it works in classify gravitational wave signals from background noises.

In Chapter 3, we will delve into the methods we have utilized in this work. We

will introduce the data preprocessing method called constant-Q transform that converts time-series data into time-frequency spectrograms. How transferring knowledge learned from models pretrained on large dataset helped in improving the deep learning models used. Moreover, backgrounds of the deep learning models are introduced to understand how each model works. How the errors of each deep learning model is presented using a loss function. Then, the performance evaluation metrics such as confusion matrix and accuracy are introduced to understand how each model is being evaluated.

In Chapter 4, we will discuss the results obtained in this work. We analyzed each result obtained from each deep learning model. The hyperparameters of each model are being optimized to obtain the best result and each model is evaluated using confusion matrix and accuracy. The differences in the results obtained from each model are discussed as well to understand why each model performed differently.

Finally, in Chapter 5 of this thesis, a conclusion is made and possible future works are discussed to improve the results of deep learning models further down the line.

CHAPTER 2

THEORY AND LITERATURE REVIEW

In this section, we will introduce the theories such as special and general relativity, black hole, gravitational waves, and deep learning that will be used in this work. We will also provide an overview of the previously published works that have influenced this dissertation.

2.1 COSMOLOGY

2.1.1 Introduction

Cosmology is a branch of astronomy that studies of the Universe as a whole and it aims to explain the origin and evolution of the Universe, from the Big Bang to today and on into the future [32]. Modern cosmology arose in the early years of the twentieth century when the great physicist, Albert Einstein introduced the principle of relativity in 1905 and general theory of relativity in 1915 that completely rewrite the laws of nature as we know them. There were other physicists throughout history that have dedicated their lives into developing relativistic cosmology by formulating new and complex language for the mathematical description of the Universe. These physicists are Wilhem de Sitter (1872–1934), Alexander Friedmann (1888–1925), and Georges Lemaître (1894–1966) [17] [33] .

One of the best models proposed by physicists to describe the origin of our Universe is the Big Bang model. This model describes that the Universe begins as a sin-

gularity 15 billion years ago where even Einstein's general theory of relativity breaks down. Space, time, matter and energy were compressed into a singular point with infinite density before expanding and created the current Universe we know filled with stars, planets, galaxies and black holes [33].

2.2 THE PRINCIPLES OF SPECIAL RELATIVITY

Albert Einstein first proposed the theory of special relativity in paper in 1905. The development of this theory corrects mechanics to handle scenarios that involve motions at a speed close to the speed of light which is known as relativistic velocities when gravitational and quantum effects are negligible [34] [35]. However, special relativity does not undermine the Newtonian model as this model is still valid and can accurately approximate motions of objects at low velocities.

2.2.1 Galilean Transformations

An inertial frame is a reference frame where in that frame, Newton's laws of motions are valid. The coordinate system attached to a body is defined as an inertial frame if it subjects to no net external force and moves in a straight line with constant velocity. It is known as Galilean invariance if Newton's laws are valid in one reference frame and in a frame moving at a uniform velocity relative to the first system [36].

Given two reference frames, S and S' where frame S' is moving at a constant velocity \mathbf{u} in the positive x -direction with respect to frame S as shown in Figure 2.1.

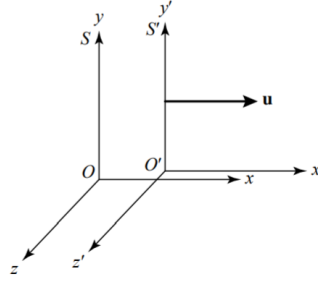


Figure 2.1: Inertial reference frames S and S' [6]

Observers in the two frames S and S' recorded the measurements and these measurements are related by the Galilean transformation equations [36] [6]:

$$x' = x - ut, \quad (2.1)$$

$$y' = y, \quad (2.2)$$

$$z' = z, \quad (2.3)$$

$$t' = t. \quad (2.4)$$

By taking the derivatives with respect to either t or t' to show how two frames are related by the components of the measured velocities \mathbf{v} and \mathbf{v}' of an object:

$$v'_x = v_x - u, \quad (2.5)$$

$$v'_y = v_y, \quad (2.6)$$

$$v'_z = v_z, \quad (2.7)$$

Equation (2.5) is the Galilean velocity transformation equation [6].

2.2.2 Einstein's Postulates and The Lorentz Transformations

Einstein proposed two postulates in his special theory of relativity paper in 1905, the first is **the principle of relativity** where all inertial reference frames obey the same laws of physics; the second is **the constancy of the speed of light** where the speed of light, c is constant, independent of the motion of its source [6]. Based on the two postulates in special relativity, Einstein derived equations called Lorentz Transformations. Referring to Figure 2.1, S' is moving at a constant velocity \mathbf{u} in the positive x -direction with respect to fixed frame S . A lightbulb goes off with the speed of light, c when $t = t' = 0$ at the origin. The speed of light is the same for both systems and the wavefronts observed must be spherical and are described by

$$x^2 + y^2 + z^2 = c^2 t^2, \quad (2.8)$$

$$x'^2 + y'^2 + z'^2 = c^2 t'^2. \quad (2.9)$$

Since the directions of y and z do not change, hence, $y = y'$ and $z = z'$. The above two equations are inconsistent with Galilean Transformations because each system will see itself to be the center of the spherical wavefront. Lorentz Transformations require both systems to be the center of the spherical wavefront where the speed of light remains unchanged [36].

Since Galilean Transformation is incorrect, the linear transformation from the standpoint of an observer in system S and an observer in system S' is

$$x' = \gamma (x - vt), \quad (2.10)$$

$$x = \gamma (x' - vt'), \quad (2.11)$$

where γ is the Lorentz factor, making the speed of light in both systems constant.

Hence, $x = ct$ and $x' = ct'$ which can be substituted into Equation (2.10) to obtain

$$ct' = \gamma(ct - vt), \quad (2.12)$$

$$ct = \gamma(ct' - vt'), \quad (2.13)$$

and by substituting Equation (2.13) into Equation (2.12), the Lorentz factor, γ is [35] [36] [6]

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}}. \quad (2.14)$$

2.3 CURVED SPACETIME OF GENERAL RELATIVITY

In 1915, Albert Einstein published yet another great work that has revolutionized modern physics: the general theory of relativity. This theory deals with theory of gravitation which explained phenomenon such as the mystery of Mercury's orbit where Newton's law of gravity failed to explain [6].

2.3.1 The Curvature of Spacetime

General relativity describes how masses warp spacetime where space and time are unified into a single four-dimensional manifold. When a massive object is present in space, it can be interpreted that space becoming curved through a fourth spatial dimension perpendicular to all of the usual three spatial directions [6].

The curvature of spacetime can be described with this analogy, imagine four people each holding a corner of the rubber sheet, being stretched tight and flat. When a basket-

ball is placed at the center of the sheet, the sheet curves downwards in response to the basketball's weight. The sheet curves through a third spatial dimension perpendicular to an original flat two-dimensional plane. Gravity is a result of spacetime curved by masses and when massless photon passes through, it will be deflected from a straight line path [36] [6].

Light always takes the shortest path, which is a straight line. However, in a curved spacetime, the shortest path light can take is a curved path called geodesic. The distance and time taken for light to travel will be longer, that is why time runs more slowly in a curved spacetime [6].

2.3.2 Principle of Equivalence and The Bending of Light

When an object on Earth is released from a certain height, it will accelerate towards the ground. All objects fall to the ground at the same time, thus weight is negligible in this scenario. The magnitude of the acceleration of an object is due to the gravitational force given by

$$ma_g = G \frac{mM}{r^2}, \quad (2.15)$$

where a_g is the acceleration and m is the mass of object. One of Einstein's postulates stated that all inertial reference frames obey the same laws of physics. However, accelerating reference frame is not an inertial frame as it produces fictitious forces that depend on acceleration [6].

The principle of equivalence states that gravitational acceleration is indistinguishable from acceleration by other forces [6] [37]. Imagine a scenario whereby a person is

in a lift dropping an apple, the apple could be falling to the ground due to gravitational acceleration. On the other hand, note that, if the lift is accelerating upwards at a rate g in free non-gravitational space, the apple would also fall to the ground as if the lift is stationary in a space with gravity. According to special relativity, inertial reference frames have a constant velocity and cannot be defined with the presence of gravity [6].

Hence, Einstein used *local reference frames*, small enough that gravitational acceleration would be constant in both magnitude and direction everywhere inside the reference frame. Essentially, gravity is abolished inside a local, freely falling reference frame [6] [37]. An experiment involving the principle of equivalence is the bending of light. Imagine a box suspended above the ground, in it there is a light source shines across the box with a straight path. When the box is released, an observer falling with the box will measure the light path to be a straight line. Whereas an observer on the ground would measure the light path to be a curved path which describes light moving in a curved path which is the quickest route through curved spacetime [6] [37].

2.3.3 Spacetime Intervals

The distance measured along the straight line between two points in a flat space that have Cartesian coordinates of (x_1, y_1, z_1) and (x_2, y_2, z_2) is defined by [6]

$$(\Delta\ell)^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2. \quad (2.16)$$

However, in Minkowski space which consists of 3 dimensional spatial coordinates and 1 dimensional spatial coordinate represented by time, we have

$$(\Delta s)^2 = -(c\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2, \quad (2.17)$$

where $(\Delta s)^2$ is the spacetime interval which is invariant under a Lorentz transformation and c is the speed of light [36]. $(\Delta s)^2$ can be positive, negative or zero which provides the causality between events given by [6]

$$(\Delta s)^2 = \begin{cases} > 0 & \text{interval is spacelike,} \\ < 0 & \text{interval is timelike,} \\ 0 & \text{interval is lightlike or null.} \end{cases} \quad (2.18)$$

2.3.4 Einstein Field Equation

The Einstein field equation describes the covariant theory of gravity, like how the geometry of spacetime is produced by the energy-momentum distribution. The equation is shown below

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}, \quad (2.19)$$

where $R_{\mu\nu}$ is the Ricci tensor, R is the Ricci scalar, $g_{\mu\nu}$ is the metric tensor, G is the gravitational constant and $T_{\mu\nu}$ is the stress-energy tensor [19].

Einstein summation is assumed for all subsequent equations whereby for any term on one side of an equation any index that is repeated, once as a superscript and once as a subscript, a summation is implied over that index. Thus, such an index is a *dummy index* that is removed by the summation and should not appear on the other side of the equation. By contracting the Riemann tensor with the metric tensor, the Ricci tensor,

$R_{\mu\nu}$ can be obtained since stress-energy tensor is a symmetric rank-2 curvature tensor

$$R_{\mu\rho\nu}^{\rho} = R_{\mu\nu}. \quad (2.20)$$

By further contracting with the metric tensor, Ricci scalar, R can be obtained.

$$g^{\mu\nu}R_{\mu\nu} = R. \quad (2.21)$$

The symmetric Einstein tensor, $G^{\mu\nu}$ can be defined as

$$G^{\mu\nu} = R^{\mu\nu} - \frac{1}{2}g^{\mu\nu}R. \quad (2.22)$$

Taking multiple contractions of Bianchi identity with the metric tensor, the result obtained is shown below

$$\nabla_{\mu}G^{\mu\nu} = 0. \quad (2.23)$$

The relation in Equation (2.23) is called contracted Bianchi identity, which indicates that Einstein tensor, $G^{\mu\nu}$ is divergenceless and it is a symmetric rank-2 curvature tensor. Thus, the covariant theory of gravity can be expressed as Equation (2.19) [19].

The stress-energy tensor or energy-momentum tensor evaluates the effect of density and flux of energy and momentum on the curvature of spacetime [19]. In the Einstein field equations, density and flux of energy and momentum are the sources of a gravitational field. The stress-energy tensor, $T^{\mu\nu}$ is of order 2 and can be represented by a 4x4 matrix

$$T^{\mu\nu} = \begin{pmatrix} T^{00} & T^{01} & T^{02} & T^{03} \\ T^{10} & T^{11} & T^{12} & T^{13} \\ T^{20} & T^{21} & T^{22} & T^{23} \\ T^{30} & T^{31} & T^{32} & T^{33} \end{pmatrix}, \quad (2.24)$$

where T^{00} is the energy density, T^{i0} is the momentum density in the direction i , T^{0i} is energy flux in the direction i , T^{ij} when $i \neq j$ is the shear stress and T^{ii} is the normal stress in the direction i where $i, j = 1, 2, 3$ [23].

The general forms of the stress-energy tensor for a perfect fluid in *flat spacetime* and *curved spacetime* that is consistent with Lorentz invariance are

$$T^{\mu\nu} = (\varepsilon + P)u^\mu u^\nu + P\eta^{\mu\nu}, \quad (2.25)$$

$$T^{\mu\nu} = (\varepsilon + P)u^\mu u^\nu + Pg^{\mu\nu}, \quad (2.26)$$

where P is the pressure, ε is the energy density, $u^\mu = \frac{dx^\mu}{d\tau}$ is the velocity 4-vector of the particles making up the perfect fluid, $\eta^{\mu\nu}$ is the Minkowski metric and $g^{\mu\nu}$ is a general metric tensor. The generalization of the conservation law to flat and curved spacetime are [19]

$$\partial_\mu T^{\mu\nu} = 0 \quad (\text{flat spacetime}), \quad (2.27)$$

$$\nabla_\mu T^{\mu\nu} = 0 \quad (\text{curved spacetime}). \quad (2.28)$$

2.3.5 The Metric for Flat Spacetime

The path between two points in a 3-dimensional space can be either a straight line path or infinitely curved paths. Hence, in order to measure a curved path between two points, a differential distance formula called a metric is used

$$(d\ell)^2 = (dx)^2 + (dy)^2 + (dz)^2, \quad (2.29)$$

$d\ell$ can be integrated along the curved path to find the distance between two points in a 3-dimensional space

$$\Delta\ell = \int_1^2 \sqrt{(d\ell)^2} = \int_1^2 \sqrt{(dx)^2 + (dy)^2 + (dz)^2}. \quad (2.30)$$

The “straightest possible line” between two points can be defined when $\Delta\ell$ is minimum. Hence, the same concept can be applied to measuring distance between two points in spacetime,

$$(ds)^2 = -(c dt)^2 + (dx)^2 + (dy)^2 + (dz)^2. \quad (2.31)$$

In this case, ds can also be integrated along the path to determine the total interval in spacetime [6],

$$\Delta s = \int_A^B \sqrt{(ds)^2} = \int_A^B \sqrt{-(c dt)^2 + (dx)^2 + (dy)^2 + (dz)^2}, \quad (2.32)$$

where A and B are two events in spacetime. A matrix representation of Minkowski metric, $\eta_{\mu\nu}$, a diagonal matrix is shown below

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.33)$$

Hence, the square of the spacetime interval, $(ds)^2$ can be represented by its matrix form as shown below [38]

$$(ds)^2 = \eta_{\mu\nu} \Delta x^\mu \Delta x^\nu = \begin{pmatrix} c \, dt & dx & dy & dz \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c \, dt \\ dx \\ dy \\ dz \end{pmatrix}. \quad (2.34)$$

2.3.6 Curved Spacetime and Schwarzschild Metric

In the presence of mass in spacetime, the curved path in that curved spacetime is called a geodesic. A geodesic is a straight worldline in a flat spacetime. In a curved spacetime, the value of the spacetime interval, Δs of a timelike geodesic can either be a minimum or a maximum called an *extremum*. In order to describe the curvature of spacetime around a massive spherical object, the spherical coordinates are used instead of Cartesian coordinates. Hence, the metric between two points in a flat space is

$$(d\ell)^2 = (dr)^2 + (r \, d\theta)^2 + (r \sin \theta \, d\phi)^2, \quad (2.35)$$

and the metric for flat spacetime is

$$(ds)^2 = -(c \, dt)^2 + (dr)^2 + (r \, d\theta)^2 + (r \sin \theta \, d\phi)^2. \quad (2.36)$$

However, spacetime will not be flat in the presence of a massive object [6].

Imagine a spherical object (planet) of mass M and radius r is placed in a coordinate system, its coordinate speed is the rate of change of its spatial coordinates. Spacetime is essentially flat at a distance ($r \simeq \infty$) from the object. Time dilation occurs when light passes through a curved spacetime. Hence, the German astronomer Karl Schwarzschild (1873–1916) obtained what is called the **Schwarzschild metric** by solving Einstein field equations two months after the general theory of relativity

was being published

$$(ds)^2 = -\left(c dt \sqrt{1 - \frac{2GM}{rc^2}}\right)^2 + \left(\frac{dr}{\sqrt{1 - \frac{2GM}{rc^2}}}\right)^2 + (r d\theta)^2 + (r \sin \theta d\phi)^2. \quad (2.37)$$

The Schwarzschild metric is only valid in the empty space outside the object, which is the spherically symmetric vacuum solution of Einstein's field equations where stress-energy tensor is zero everywhere. The proper distance or radial distance between two points measured simultaneously ($dt = 0$) and on the same radial line ($d\theta = d\phi = 0$) is

$$d\mathcal{L} = \frac{dr}{\sqrt{1 - \frac{2GM}{rc^2}}}. \quad (2.38)$$

Likewise, the proper time, $d\tau$ is related to the time dt by setting $dr = d\theta = d\phi = 0$ at an infinite distance by

$$d\tau = \sqrt{1 - \frac{2GM}{rc^2}} dt, \quad (2.39)$$

time passes more slowly on a planet since $d\tau < dt$ [6].

2.4 BLACK HOLES

When a star, more massive than the Sun has exhausted its sources of nuclear fusion, gravitational collapse occurs where the massive star gets crushed by its own gravity, a black hole is formed. The gravitational field of black hole is extremely strong that not even light can escape its pull [6].

Black holes come in a range of masses, such as stellar-mass black holes with masses ranging from 3 to 15 M_{\odot} and intermediate-mass black holes (IMBHs) with masses ranging from 100 to 1000 M_{\odot} . At the center of most galaxies in this Universe

lie supermassive black holes (SMBH) with masses ranging from 10^5 to $10^9 M_\odot$. Black holes in the earliest instants of the Universe are called primordial black holes with masses ranging from 10^{-8}kg to $10^5 M_\odot$ [6].

2.4.1 The Schwarzschild Radius

The Schwarzschild radius is a physical parameter that defines the radius of the event horizon of a Schwarzschild black hole. A black hole is formed when a massive star shrinks to its Schwarzschild radius and is given by

$$R_s = \frac{2GM}{c^2}, \quad (2.40)$$

where G is the gravitational constant, M is mass of object and c is the speed of light. At the Schwarzschild radius, $d\tau = 0$ and it shows that time has stopped when measured by an observer with a great distance away [6].

If an observer were to measure the speed of light above the collapsed star, it will always be c . The rate of change of spatial coordinates of a photon is called the *co-ordinate speed of light* and it can be calculated using Schwarzschild metric by setting $ds = 0$,

$$0 = -\left(c dt \sqrt{1 - \frac{2GM}{rc^2}}\right)^2 + \left(\frac{dr}{\sqrt{1 - \frac{2GM}{rc^2}}}\right)^2 + (r d\theta)^2 + (r \sin \theta d\phi)^2. \quad (2.41)$$

By inserting $d\theta = d\phi = 0$, the coordinate speed of light is given by

$$\frac{dr}{dt} = c \left(1 - \frac{2GM}{rc^2}\right) = c \left(1 - \frac{R_s}{r}\right), \quad (2.42)$$

when $r \gg R_s$, $\frac{dr}{dt} \simeq c$ which is in flat spacetime. However, $\frac{dr}{dt} = 0$ when $r = R_s$, which means light is frozen in time t at Schwarzschild radius, information cannot be accessed from within the Schwarzschild radius [6].

A black hole is enclosed by an **event horizon** where once entered, there is no return, not even light can escape. This is where time stopped and things seem to be frozen from the observer's view at a great distance away. At the center of a black hole, is the **singularity**, a point of zero volume and infinite density and the spacetime curvature is also infinite for a nonrotating black hole [6].

2.4.2 Journey into a Black Hole

Imagine an observer positioned at a great distance away and reflecting a radio wave from an object at the event horizon of a black hole, the trip is symmetric. The time taken for a radio photon to reach the event horizon at a radial distance $r \gg R_s$ and return to the observer is

$$\begin{aligned}\Delta t &= \int_{r_1}^{r_2} \frac{dr}{\frac{dr}{dt}}, \\ &= \int_{r_1}^{r_2} \frac{dr}{c \left(1 - \frac{R_s}{r}\right)}, \\ \Delta t &= \frac{r_2 - r_1}{c} + \frac{R_s}{c} \ln \frac{r_2 - R_s}{r_1 - R_s},\end{aligned}\tag{2.43}$$

assuming $r_1 < r_2$. If $r_1 = R_s$, $\Delta t = \infty$. According to gravitational time dilation, the radio photon's coordinate velocity will slow down until it stops with a time taken of infinity [6].

Imagine an indestructible space probe approaches a black hole with a mass of 10

M_{\odot} . The space probe flashes a monochromatic light every second to an observer at a great distance away. As the space probe keeps approaching, the light signals arrive further and further apart due to gravitational time dilation and the coordinate speed of light becomes slower. The frequency of the light waves get redshifted due to gravitational redshift and the energy of the photon decreases with time. As the fall progresses, the side of the probe that faces the black hole starts to experience stronger gravitational pull. The probe is being stretched and the phenomenon is called **spaghettification** [6] [39].

According to the Schwarzschild metric, a particle cannot be at rest when $r > R_s$ and by using $dr = d\theta = d\phi = 0$, the interval is given by

$$(ds)^2 = -(c dt)^2 \left(1 - \frac{R_s}{r}\right) > 0, \quad (2.44)$$

the interval is *spacelike* when $r < R_s$, which is impossible to be at rest. The probe is able to record the light from the events in the outside Universe falling behind it. After a certain proper time, the probe passes the event horizon and is drawn to the singularity [6].

2.5 GRAVITATIONAL WAVES

Maxwell's theory predicts that electromagnetic waves are ripples of electric and magnetic forces perpendicular to each other spreading out through the vacuum of space at the speed of light. Hence, Einstein's general theory of relativity would predict the same thing but for gravitation. Such a wave is called a **gravitational wave**, created by accelerating masses in space such as a binary neutron star or a black hole system [19]

[20].

It is not an easy task to detect such a wave as the magnitude of the wave is extremely small when it reaches Earth. However, thanks to the advancement of modern technology, physicists were able to detect gravitational waves with more sensitive equipment and have proven its existence, 100 years after Einstein predicted them in his general theory of relativity [18] [19].

2.5.1 A Linearized Gravitational Wave

Gravitational waves are ripples or small perturbations propagating through the flat spacetime metric. The metric for flat spacetime is $g_{\mu\nu} = \eta_{\mu\nu}$, where $\eta_{\mu\nu}$ is Minkowski metric and shown in Equation (2.33). Since gravitational waves are assumed to be weak perturbations, the metric of spacetime close to flat with small perturbations can be expressed as

$$g_{\mu\nu}(x) = \eta_{\mu\nu} + h_{\mu\nu}(x), \quad (2.45)$$

where $h_{\mu\nu}(x)$ is the small perturbations that represent gravitational waves propagating through a close to flat spacetime [1]. The value of $h_{\mu\nu}(x)$ is small, it can be assumed that $||h_{\mu\nu}|| \ll 1$ and the square of it, $h_{\mu\nu}h_{\alpha\beta} \approx 0$ [19] [1].

Let's consider the transformation of these components under an n infinitesimal coordinate transformation at a point S,

$$g_{\sigma'\rho'} = g_{\mu\nu} \frac{\partial x^\mu}{\partial x^{\rho'}} \frac{\partial x^\nu}{\partial x^{\sigma'}}, \quad (2.46)$$

$$x^\mu(S) = x^\mu(S) + \xi^\mu(S), \quad (2.47)$$

where $|\xi^\mu| \ll |x^\mu|$,

$$g_{\sigma'\rho'}(x) = \frac{\partial x^\mu}{\partial x^{\rho'}} \frac{\partial x^\nu}{\partial x^{\sigma'}} g_{\mu\nu}(x^\mu - \xi^\mu). \quad (2.48)$$

Calculating $h_{\mu\nu}$, ξ^μ and their derivatives to first order,

$$\frac{\partial x^\mu}{\partial x^\rho} = \delta^\mu_\rho - \frac{\partial \xi^\mu}{\partial x^\rho}, \quad (2.49)$$

$$g_{\mu\nu}(x^\mu - \xi^\mu) = \eta_{\mu\nu} + h_{\mu\nu}. \quad (2.50)$$

Apply the concept of Equation (2.49) on $\frac{\partial x^{\nu'}}{\partial x^{\sigma'}}$, then substitute into Equation (2.46) along with Equations (2.49) and (2.50)

$$\frac{\partial x^\mu}{\partial x^{\rho'}} = \delta^\mu_\rho - \frac{\partial \xi^\mu}{\partial x^{\rho'}}, \quad (2.51)$$

$$g_{\mu\nu}(x^{\mu'} - \xi^\mu) = \eta_{\mu\nu} + h_{\mu\nu}, \quad (2.52)$$

gives the following equation up to first order

$$\begin{aligned} g_{\sigma'\rho'} &= (\delta^\mu_\rho - \frac{\partial \xi^\mu}{\partial x^{\rho'}})(\delta^\nu_\sigma - \frac{\partial \xi^\nu}{\partial x^{\sigma'}})(\eta_{\mu\nu} + h_{\mu\nu}) \\ &\approx \eta_{\sigma\rho} + h_{\sigma\rho} - \partial_\sigma \xi_\rho - \partial_\rho \xi_\sigma. \end{aligned} \quad (2.53)$$

By substituting $g_{\sigma'\rho'} = \eta_{\sigma\rho} + h_{\sigma'\rho'}$ into Equation (2.53), metric of the same form but with new perturbation is given by

$$h_{\sigma'\rho'} = h_{\sigma\rho} - \partial_\sigma \xi_\rho - \partial_\rho \xi_\sigma. \quad (2.54)$$

This transformation in the linear field approximation is called gauge transformation, which leads to new potentials but same fields [1].

The products of Christoffel symbols in the Riemann tensor can be neglected since

it is to first order $h_{\mu\nu}$, the Riemann tensor and Christoffel symbol are given by

$$R_{\alpha\mu\beta\nu} = \partial_\beta \Gamma_{\alpha\mu\nu} - \partial_\nu \Gamma_{\alpha\mu\beta}, \quad (2.55)$$

$$\Gamma_{\alpha\mu\nu} = \frac{1}{2}(\partial_\nu h_{\mu\alpha} + \partial_\mu h_{\nu\alpha} - \partial_\alpha h_{\mu\nu}). \quad (2.56)$$

Hence, the new equation for the Riemann tensor is

$$R_{\alpha\mu\beta\nu} = \frac{1}{2}(\partial_\mu \partial_\beta h_{\nu\alpha} + \partial_\alpha \partial_\nu h_{\mu\beta} - \partial_\alpha \partial_\beta h_{\mu\nu} - \partial_\mu \partial_\nu h_{\alpha\beta}), \quad (2.57)$$

and the Ricci tensor to first order is thus

$$R_{\mu\nu} = \frac{1}{2}(\partial_\alpha \partial_\mu h^\alpha_\nu + \partial_\alpha \partial_\nu h^\alpha_\mu - \partial_\mu \partial_\nu h - \square h_{\mu\nu}), \quad (2.58)$$

where $\square \equiv \eta^{\alpha\beta} \partial_\alpha \partial_\beta = \partial_\alpha \partial^\alpha$ and $h \equiv h^\alpha_\alpha = \eta^{\alpha\beta} h_{\alpha\beta}$ respectively [19] [1] [38]. The Ricci scalar obtained is

$$R = \partial_\mu \partial_\nu h_{\mu\nu} - \square h. \quad (2.59)$$

Hence, the linearized Einstein tensor and the field equations obtained by substituting equations above are given by

$$E_{\mu\nu} = \frac{1}{2}[\partial_\alpha \partial_\mu h^\alpha_\nu + \partial_\alpha \partial_\nu h^\alpha_\mu - \partial_\mu \partial_\nu h - \square h_{\mu\nu} - \eta_{\mu\nu}(\partial_\alpha \partial_\beta h_{\alpha\beta} - \square h)], \quad (2.60)$$

$$\partial_\alpha \partial_\mu h^\alpha_\nu + \partial_\alpha \partial_\nu h^\alpha_\mu - \partial_\mu \partial_\nu h - \square h_{\mu\nu} - \eta_{\mu\nu}(\partial_\alpha \partial_\beta h_{\alpha\beta} - \square h) = \frac{16\pi G}{c^4} T_{\mu\nu}. \quad (2.61)$$

In order to simplify Equation (2.61), the following equation is introduced

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h. \quad (2.62)$$

Thus, the simplified version of Equation (2.61) is given by

$$\partial_\alpha \partial_\mu \bar{h}_\nu^\alpha + \partial_\alpha \partial_\nu \bar{h}_\mu^\alpha - \square \bar{h}_{\mu\nu} - \eta_{\mu\nu} \partial_\alpha \partial_\beta \bar{h}_{\alpha\beta} = \frac{16\pi G}{c^4} T_{\mu\nu}. \quad (2.63)$$

Gauge transformation is performed in order to simplify further the equations, the transformed metric $\bar{h}'_{\alpha\beta}$ is given by

$$\bar{h}'_{\alpha\beta} = \bar{h}_{\alpha\beta} - \partial_\beta \xi_\alpha - \partial_\alpha \xi_\beta + \eta_{\alpha\beta} \partial_\sigma \xi^\sigma. \quad (2.64)$$

The transformed divergence of $\bar{h}_{\alpha\beta}$ then becomes

$$\partial_\beta \bar{h}'^\beta_\alpha = \partial_\beta \bar{h}^\beta_\alpha - \square \xi_\alpha, \quad (2.65)$$

by choosing ξ_α such that $\xi_\alpha = \partial_\beta \bar{h}^\beta_\alpha$ and dropping the prime, the equation obtained is

$$\partial_\beta \bar{h}^\beta_\alpha = 0. \quad (2.66)$$

This is called the Lorenz gauge or Lorenz condition, which is a class of coordinate systems. The field equation is reduced to

$$\square h_{\mu\nu} = -\frac{16\pi G}{c^4} T_{\mu\nu}. \quad (2.67)$$

Coordinates that obey the Lorenz condition are called *harmonic*, it is given by

$$V_\alpha = \partial_\beta h^\beta_\alpha - \frac{1}{2} \partial_\alpha h^\beta_\beta = 0. \quad (2.68)$$

The field equations of empty space in a time-dependent case are

$$\square h_{\mu\nu} = 0, \quad (2.69)$$

which is d'Alemberts wave equation. It shows that gravitational waves move in empty space with the speed of light. The corresponding Riemann tensor equation is [19] [1] [38]

$$\square R_{\alpha\mu\beta\nu} = 0. \quad (2.70)$$

2.5.2 Detecting Gravitational Waves

Gravitational waves cannot be detected locally due to principle of equivalence where the effect of gravity is indistinguishable from ordinary acceleration effect. Hence, using only one test particle is not applicable to detect gravitational waves. Instead, two or more test particles at different locations are being used to detect gravitational waves [1].

Consider a gravitational wave propagating in the z- direction, two test masses A and B are at rest with their coordinates being $x_A^i = (0, 0, 0)$ and $x_B^i = (x_B, y_B, z_B)$, respectively. The initial four-velocities of both test masses are $u_A = u_B = (1, 0, 0, 0)$. The geodesic equation must be solved for each particle in order to predict the motion after the wave hits. The geodesic equation is given by

$$\frac{d^2 x^i}{d\tau^2} = -\Gamma_{\alpha\beta}^i \frac{dx^\alpha}{d\tau} \frac{dx^\beta}{d\tau}, \quad (2.71)$$

this equation is being evaluated only to first order to calculate the first-order changes $\delta x^i(\tau)$. $\Gamma_{\alpha\beta}^i$ vanishes in the flat spacetime and one will find

$$\frac{d^2 \delta x^i}{d\tau^2} = -\delta \Gamma_{\alpha\beta}^i u^\alpha u^\beta = -\delta \Gamma_{tt}^i, \quad (2.72)$$

where the u^α are the unperturbed four-velocities and $\delta \Gamma_{\alpha\beta}^i$ are the first-order changes in the Christoffel symbols. $\delta \Gamma_{tt}^i$ is easily evaluated as it vanishes along with $\delta \Gamma_{\alpha\beta}^i$. Thus,

$$\frac{d^2 \delta x^i}{d\tau^2} = 0. \quad (2.73)$$

The test masses A and B are initially at rest and $\delta x^i = 0$, so $\frac{d^2 \delta x^i}{d\tau^2} = 0$. Which implies that for all τ , $\delta x^i = 0$ for both masses

$$\delta x_A^i = \delta x_B^i = 0. \quad (2.74)$$

Therefore, as the waves pass through in the amplitude of the waves, the coordinate positions of both masses remain unchanged. However, the distance between both masses do change with time due to both masses oscillate with the frequency of the waves [1].

Consider the exact situation given above, both masses are positioned at a distance L away from each other along the x -axis in an unperturbed flat spacetime. When gravitational waves pass through, the distance between both masses $L(t)$ measured along the x -axis is

$$L(t) = \int_0^L \sqrt{dx[1 + h_{xx}(t, 0)]} \approx L[1 + \frac{1}{2}h_{xx}(t, 0)], \quad (2.75)$$

where the change in distance along the x -axis is given by

$$\frac{\delta L(t)}{L} = \frac{1}{2}h_{xx}(t, 0). \quad (2.76)$$

The distance between both masses does change to the variation of time of the wave as shown in Figure 2.2. If the wave has frequency ω , amplitude a , and phase δ , the fractional change in the distance is

$$\frac{\delta L(t)}{L} = \frac{1}{2}a \sin(\omega t + \delta), \quad (2.77)$$

where it is half of the amplitude of the wave that oscillates periodically along the x -axis. This equation is generalized to the case where one test mass is at the origin whereas the other is at an arbitrary location. Suppose the second test mass is at L distance away in the direction of unit vector \vec{n} where plane $z = 0$. The distance $L(t)$ measured along the path as the flat space changes is

$$\frac{\delta L(t)}{L} = \frac{1}{2}h_{ij}(t,0)n^i n^j, \quad (2.78)$$

the fractional change in distance, $\frac{\delta L(t)}{L}$ is the fractional strain produced by the gravitational wave [1].

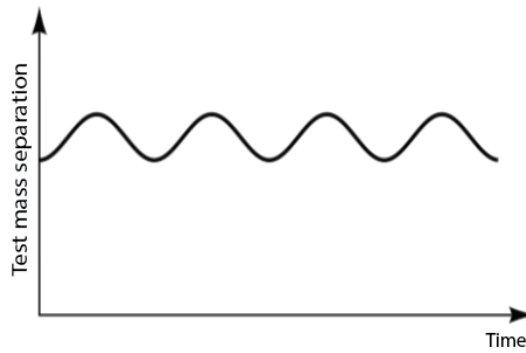


Figure 2.2: Distance separation of two test masses perturbed by a gravitational wave [1]

2.5.3 Gravitational Wave Polarization

Similar to electromagnetic waves, gravitational waves are transverse waves where test masses in the transverse directions (x and y directions unless the wave is incline along the z direction) will be affected. There are two types of polarized gravitational waves, plus-polarized and cross-polarized [1] [19] [38].

Let's consider a plane wave, $h_{ab} = \epsilon_{ab} e^{ikx}$ where the symmetric matrix, ϵ_{ab} is called the polarization tensor. The Lorenz condition, $\partial_\beta \bar{h}_\alpha^\beta = 0$ and the wave equation constrain its ten independent components. Since h_{ab} satisfies the wave equation and by adding four function ξ_a with $\square \xi_a$, the four components of h_{ab} will vanish identically. By setting $\alpha = 1, 2, 3$ in the *transverse traceless* (TT) gauge,

$$h_{0\alpha} = 0, \quad (2.79)$$

$$h_\alpha^\alpha = h = 0. \quad (2.80)$$

Hence, the harmonic gauge condition from Equation (2.68) becomes

$$V_\alpha = \partial_\beta h_\alpha^\beta, \quad (2.81)$$

$$V_0 = \partial_\beta h_0^\beta = \partial_0 h_0^0 = i\omega \epsilon_{00} e^{ikx} = 0, \quad (2.82)$$

$$V_\mu = \partial_\beta h_\mu^\beta = ik^\beta \epsilon_{\mu\beta} e^{ikx} = 0. \quad (2.83)$$

Thus, $\epsilon_{00} = 0$ and the polarization tensor is transverse, $k^\beta \epsilon_{\mu\beta} = 0$ [19] [23].

The metric perturbation h_{ab} can transform under rotations of the coordinate system. Helicity is the projection of the spin on the direction of momentum. A gravitational

plane wave can be decomposed into three helicity components, ± 2 , ± 1 and 0 . However, the helicity components, ± 1 and 0 vanish with suitable choice of coordinates, leaving only helicity components of ± 2 . This shows that gravity may be associated with spin-2 massless fields [19].

Assume that gravitational wave propagating along the z -axis with fixed frequency ω is chosen, general solution of the linearized Einstein equations in transverse–traceless (TT) gauge is

$$h_{ab}(t, z) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \epsilon_{11} & \epsilon_{12} & 0 \\ 0 & \epsilon_{12} & -\epsilon_{11} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} e^{i\omega(z-t)}. \quad (2.84)$$

The part of the wave that is ϵ_{xx} is called the *plus polarization*, denoted by $+$. Whereas the other part, $\epsilon_{xy} = \epsilon_{yx}$ is called the *cross polarization*, denoted by \times [19].

The generalized form of gravitational wave propagating along the z -axis with fixed frequency ω is

$$h_{ab}(t, z) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & f_+(t-z) & f_\times(t-z) & 0 \\ 0 & f_\times(t-z) & -f_+(t-z) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.85)$$

where $f_+(t-z)$ and $f_\times(t-z)$ are dimensionless functions. These functions characterize the shape and amplitude of the wave [19]. The effect of purely plus-polarized and purely cross-polarized gravitational waves on a circular array of test masses in its initial state are shown in Figure 2.3. The gravitational wave stretches one direction and squeezes the other since gravitational wave is a tidal field. Each polarization leads to an elliptical oscillation with cross-polarization ellipse rotated by 45° relative to plus-polarization ellipse [7].

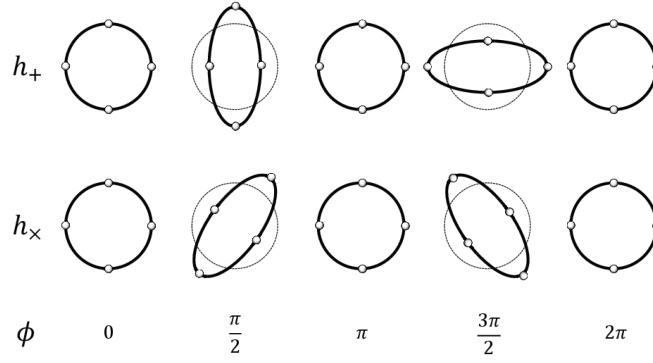


Figure 2.3: Effect of purely plus-polarized and purely cross-polarized gravitational waves on a circular array of test masses [7]

2.5.4 Power and Energy in Gravitational Waves

In a Newtonian gravitational field, the energy density is given by

$$\varepsilon(x) = -\frac{1}{8\pi G}[\nabla\Phi(x)]^2, \quad (2.86)$$

where $\nabla\Phi(x)$ is the Newtonian gravitational potential. In general relativity, there is no local energy density. It would contradict the principle of equivalence if such energy density exists as stated in the principle, gravity is abolished in a local reference frame [1]. However, there is an association between energy density and a weak gravitational wave of wavelength λ , provided that λ is shorter than the scale of curvature R of the background spacetime in which the wave propagates through. This energy is not local, instead it is the average energy density over spacetime volumes, larger than λ but smaller than R [19].

The approximation of this energy density becomes more accurate at large distances from the gravitational wave source where $\frac{\lambda}{R} \rightarrow 0$. At great distances away from the source, space is nearly flat and $R \rightarrow \infty$. The power emitted by gravitational waves is

given by

$$P(t, r) = \frac{G}{5c^2} \langle \ddot{Q}_{ij} \ddot{Q}^{ij} \rangle, \quad (2.87)$$

where the triple dot signifies third time derivative, $\langle \rangle$ denotes a time average over a wave period and Q is the reduced quadrupole tensor which is defined by

$$Q_{ij} = q_{ij} - \frac{1}{3} \delta_{ij} q^k_k, \quad (2.88)$$

where q_{ij} is the second mass momentum [19].

2.5.5 Gravitational versus Electromagnetic Waves

Electromagnetic waves interact strongly with matter whereas gravitational waves interact extremely weakly due to the fact that gravitational waves are more difficult to be detected than electromagnetic waves and gravitational waves do not get absorbed by matter like electromagnetic waves do. Electromagnetic waves have smaller wavelengths than gravitational waves which can be used to form images whereas gravitational waves cannot. Thus, the technique to detect gravitational waves is likened to hearing sound [19].

Photons emitted from an astronomical source is usually incoherent since they are generated by independent events involving atoms, ions or electrons whereas gravitons are phase-coherent when emitted because they are generated by bulk motion of matter or spacetime curvature which is analogous to laser light. There are two observational consequences from this coherence. (1) The matched filtering techniques can extend the distance at which sources can be detected by a factor of roughly \sqrt{n} where n is the number of cycles observed for the wave if the waveform is well modeled. (2) The

strain of gravitational wave falls off as $\frac{1}{r}$ at distance r from a source as compared to the incoherent electromagnetic waves where the energy flux falls off as $\frac{1}{r^2}$ [19].

The differences between gravitational waves and electromagnetic waves are illustrated in Table 2.1

Table 2.1: Differences between gravitational waves and electromagnetic waves [1].

	Gravitational Wave	Electromagnetic Wave
Field equation	Einstein equation $g_{\mu\nu}(x) = \eta^{\mu\nu} + h_{\mu\nu}$	Maxwell's equations $\nabla \cdot E = \frac{\rho}{\epsilon_0}$ (Gauss' law) $\nabla \cdot B = 0$ (Gauss' law for magnetism) $\nabla \times E = -\frac{\partial B}{\partial t}$ (Faraday's law) $\nabla \times H = J + \frac{\partial D}{\partial t}$ (Ampere-Maxwell law)
Basic potentials	Linearized metric perturbations $h_{\mu\nu}(x)$	Vector and scalar potentials $(\Phi(x), \vec{A}(x))$
Sources	Stress-energy $T_{\mu\nu}$	Charge and current $(\rho_{electric}, \vec{J})$
Gauge transformation	$h_{\sigma'\rho'} = h_{\sigma\rho} - \partial_{\sigma}\xi_{\rho} - \partial_{\rho}\xi_{\sigma}$	$\vec{A} = \vec{A} + \nabla\Lambda$ $\Phi = \Phi - \frac{\partial\Lambda}{\partial t}$
Example of gauge condition	Lorenz gauge $\partial_{\beta}h_{\alpha}^{\beta} - \frac{1}{2}\partial_{\alpha}h_{\beta}^{\beta}$	Lorenz condition $\vec{\nabla} \cdot \vec{A} = \frac{\partial\Phi}{\partial t} = 0$
Field equations simplified by the gauge condition	$\square h_{\mu\nu} = 0$	$\square \vec{A} = 0$ $\square \Phi = 0$

2.5.6 Gravitational Wave Interferometers

A number of gravitational wave detectors around the world are based on the principle of the Michelson interferometer shown in Figure 2.4.

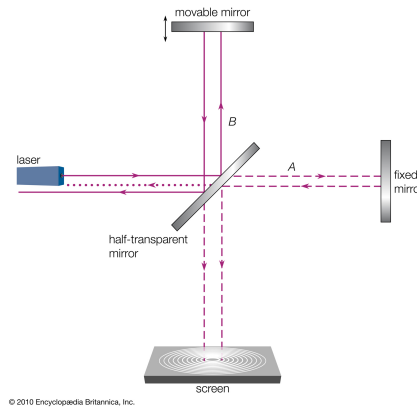


Figure 2.4: Setup of Michelson–Morley Experiment [8]

A beam of light from a laser splits into two beams running along two perpendicular arms of the interferometer that are about several kilometres long. These two beams then get reflected back to the origin to recombine and interfere with each other [1]. In physics, interference is a phenomenon where two waves superimpose to form a resultant wave with either greater, smaller or zero amplitude.

The principle of superposition of waves states that when two or more waves overlap, the resultant amplitude at any point and instant is found to be the addition of the amplitudes of the waves. Constructive interference occurs when the resultant amplitude of the waves is greater than the amplitudes of the input waves, whereas destructive interference occurs when the resultant amplitude of the waves is either zero or less than the amplitudes of the input waves. The detection of gravitational waves is based on the theory mentioned above. When gravitational waves pass through the detector, the dis-

tance will increase and decrease periodically causing either constructive or destructive interference to occur. The equations of constructive and destructive interference are given as

$$\Delta L \equiv L_A - L_B = n\lambda \quad (\text{Constructive interference}), \quad (2.89)$$

$$\Delta L \equiv L_A - L_B = (n + \frac{1}{2})\lambda \quad (\text{Destructive interference}), \quad (2.90)$$

where $n = 0, 1, 2, \dots$, and L_A and L_B are the lengths of two arms in the interferometer. The arm lengths will expand and contract periodically. However, the same condition will not happen at the same time for both arms. If gravitational waves pass through, one will expand while the other will contract and vice versa [1]. There are in total of three LIGO (Laser Interferometer Gravitational-Wave Observatory) gravitational wave detectors, located at Hanford, Washington and in Livingston, Louisiana [18]. Figure 2.5 shows the LIGO gravitational wave detector at Hanford, Washington.



Figure 2.5: LIGO gravitational wave detector at Hanford, Washington [9].

2.6 DEEP LEARNING

With the advancement of modern technologies, more advanced computing methods are

demanding which gives rise to the development of artificial intelligence, machine learning and deep learning. Companies use such advanced computing methods to improve product revenue and identify customer behaviours whereas in the field of science, such methods can be used to recognize a certain patterns in the data and classify them based on these patterns [12].

Artificial intelligence is intelligence simulated by computer systems which requires more complicated and advanced computing methods to develop. Machine learning is a subfield of artificial intelligence which is a combination of mathematics, statistics and computer science. Machine learning refers to the study of computer algorithms that will improve its behaviour through gaining experience by learning from massive amount of data. Deep learning is a subfield of machine learning that builds algorithms and use multiple layers progressively to learn high and low level of abstractions of data. Deep learning models are usually applied in fields such as computer vision, medical image analysis and speech recognition [12] [40].

2.6.1 Convolutional Neural Network(CNN)

Convolutional neural network is a class of artificial neural network and is designed to be used for computer vision and to analyze visual imagery which uses multiple layers to calculate an output of a given dataset. The design of CNN is similar to the structure of the animal visual cortex as described by Hubel and Weisel [12]. Figure 2.6 shows the architecture of a CNN.

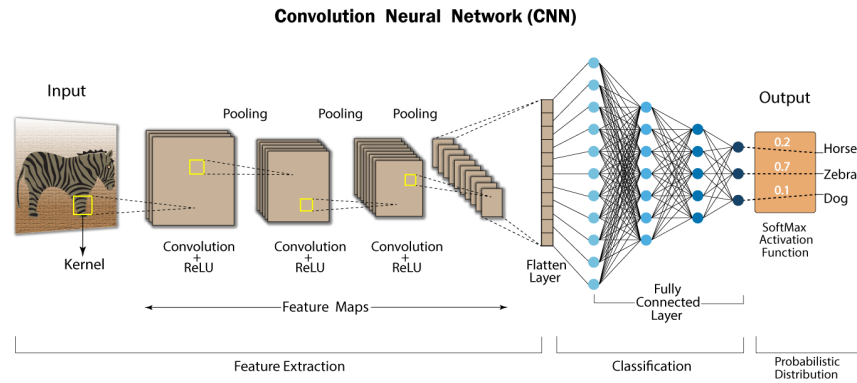


Figure 2.6: Architecture of CNN [10].

Each block represents different layers and there are in total of three main layers, the input, hidden and fully connected layers. The model will output the final result after the final layer. When dealing with a black and white image, each pixel is represented as an entry in a matrix with 0 and 1 whereas the value will be either 1 or 256 for black and white in color images [12].

The **convolutional layer** is the first layer after passing through the input image and it is where majority of the computations in CNN occur. There are filters in this layer that scan over a portion of the image. Every filter extends through the entirety of the length of this layer but not particularly large with respect to height and width. When an image is inputted, white pixels will be distinguished as having a value of 1 while black pixels will be distinguished as having a value of -1. The model will then extract the unique features of an image, usually colors, shapes, and edges that define a specific image. Once features of a given training image are obtained, filtering will be performed over this inputted image. In the filtering process, let's say a 3×3 pixel square is taken from an image feature to match with a patch of that 3×3 pixel square inputted image [12].

The number of the pixels of the feature is then multiplied with the number of pixels of the image patch. When the pixels match perfectly, an output of 1 will be obtained whereas an output of -1 will be obtained when the pixels do not match perfectly. Then, the average of the pixel products is taken, the average should be 1 if the image matches exactly or else degrees lower than 1 will be obtained. The average value is placed in the center of the position of the image patch with a given feature on what is called a feature map which will be the output of the convolution layer and be used for the next layer. The convolution layer will produce multiple feature maps over different iterations and this process is known as convolving an image. The feature or activation map is denoted by

$$h_{i,j}^k = \tanh((w^k x)_{i,j} + b_k), \quad (2.91)$$

where b_k is the bias, x is the value of specific pixel, w^k is the weight and \tanh , the hyperbolic tangent function used to introduce non-linearities in the data. i and j subscripts refer to the entry of the matrix that represents the feature or activation map. The pixels in the feature maps are connected to the preceding layer by the weight, w^k . The feature maps are put into the pooling layer and the spatial size of the output volume is calculated as

$$Spatial\ Size_{Output} = \frac{W - F + 2P}{S + 1}, \quad (2.92)$$

where F is the size of receptive field in the convolution layer, P is the amount of padding, S is the stride and W is the input volume size [12].

The **pooling layer** is the layer between successive convolutional layers. This layer "pools" feature maps produced in the convolution layer into an image. The pooling

layer performs dimensionality reduction effectively and thereby reduce the complexity of the model which helps to prevent overfitting. If the pooling layer has a 2×2 filter and a stride of 1, when the filter is applied to the feature maps, it will move across the height and width by 1 pixel. Max pooling is when the maximum value inside the filter is chosen when moving across a $m \times n$ feature map [12].

The **Rectified Linear Units (ReLU)** layer is the layer that applies this function

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0, \\ 0, & \text{if } x_i < 0, \end{cases} \quad (2.93)$$

to the inputs to this layer where x is the input to a neuron. When the function is applied, the negative values of the feature map will be zeroed [12] [41]. Figure 2.7 shows the graph of ReLU activation function [11].

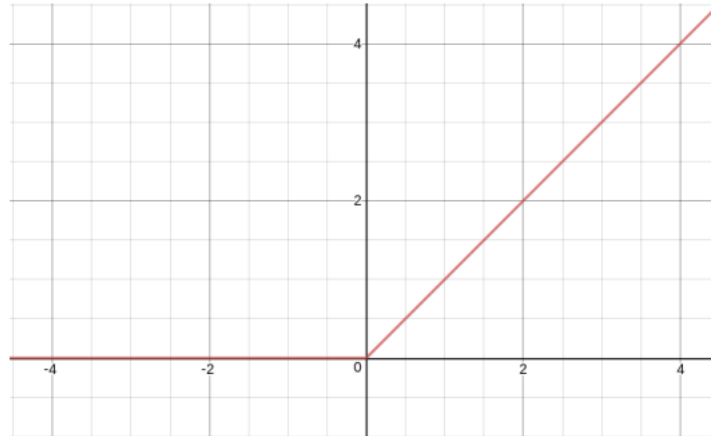


Figure 2.7: Graph of ReLU activation function [11].

The **sigmoid layer**, also refers to as the logistic function is a non-linear activation function used mostly in feedforward neural networks [41].

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (2.94)$$

It follows the principle where lower influence produces low output and higher influence produces higher output, which renders the output between 0 and 1 [12]. It is used for predicting a probability-based output and can be applied in modeling logistic regression tasks and binary classification problems [41]. Figure 2.8 shows the graph of sigmoid activation function.

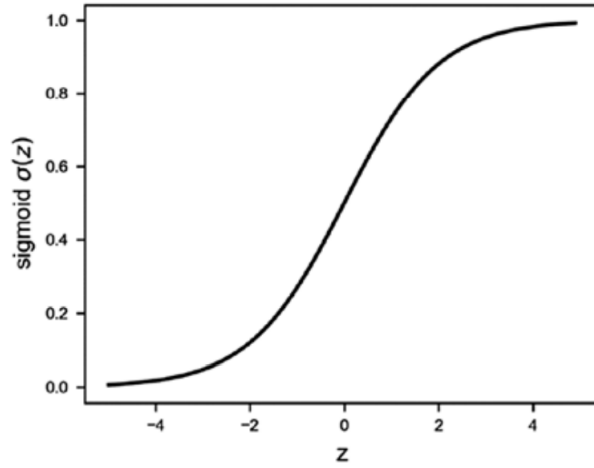


Figure 2.8: Graph of sigmoid activation function [12].

Both ReLU and sigmoid layers are used in dense layers to determine which neurons should be activated or deactivated.

The **fully connected (FC) layer** is where all neurons are connected to the activation maps in the preceding layer. This layer is placed after the convolutional, pooling, and ReLU layers. The images inputted to this layer are reduced due to prior operations and are scanned, which correspond to each feature map, each of the values given are turned into a list of values. The final classification is done here which shows the probability

of the image being a 1 or a 0 [12].

The **loss layer** is the layer where the predicted labels from the actual labels of the images are compared. A *Softmax loss classifier* is used to classify objects from k possible feature levels, while the *Euclidean function* is used for regressing against the labels of the specific images. The functions are given by

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (\text{Softmax loss classifier}), \quad (2.95)$$

$$E = \frac{1}{2N} \sum_{i=1}^N \|\hat{y}_i - y_i\|_2^2 \quad (\text{Euclidean loss function}). \quad (2.96)$$

During backpropagation, the loss calculated from the output layer is passed to the input layer and the weight of each neuron can be adjusted accordingly. An optimizer function is used to determine the degree to which the weights should change upon each iteration. By using the derivatives, partial derivatives, and the chain rule in calculus to find its scalar value, the gradient is then subtracted which will lead to a solution that is more optimal until a global optimum is reached. Hence, the optimizer function can be used to reduce the loss function by making small changes to the weight parameters and thus, the network learns to make accurate predictions for the given training samples. There are two main optimizer functions which will be discussed in this section: the stochastic gradient descent (SGD) and Adam optimizer functions [12] [40].

Stochastic Gradient Descent (SGD) calculates the loss and updates the weight after the pass of every training sample in each iteration. The formula for weight updates can be expressed as follows:

$$Weights = Weights - learning\ rate * Loss, \quad (2.97)$$

where learning rate is a parameter defined in the network, having a typical value of 0.01. The number of iterations can be reduced by providing a minibatch, the high fluctuations in the SGD optimizations can be reduced. It enables averaging the loss for all samples in a batch and updating the weights at the end of the batch which results in a smoother training process [40].

Furthermore, the Adam optimizer function, also known as Adaptive Moment Estimation is by far the most popular widely used optimizer function in deep learning. This optimization technique computes an adaptive learning rate for each parameter. It defines momentum and variance of the gradient of the loss and leverages a combined effect to update the weight parameters. Both momentum and variance help smoothen the learning curve and effectively improve the learning process. The formula can be expressed as follows:

$$Weights = Weights - (Momentum\ and\ Variance\ combined), \quad (2.98)$$

where the default values of Adam optimizer function work quite effectively and do not need to be changed for most use cases [40].

Overfitting occurs when multi-layer perceptrons have more than one layer, hence, it is recommended to regularize the input data. DropOut is one of the methods where a probability is set as a hyperparameter, making certain hidden layers not pass through based on this probability. The DropOut method has been proven to be effective as

it improves the computational efficiency of networks with large amounts of parameters. It causes a network to become smaller during a given iteration. Moreover, L2 regularization method tends to penalize abnormally large weights and favor those that are generally mild in their proportion relative to the entirety of the matrix [12]. In L2 regularization, the squared weights are added to the loss function and the values of the weights are reduced to near 0. L2 is highly recommended over L1 for reducing overfitting in most cases. The equation can be represented as

$$Cost\ Function = Loss + \frac{\lambda}{2m} * \| Weights \|^2, \quad (2.99)$$

and it is also called the "weight decay" [40].

2.7 LITERATURE REVIEW

There are other researchers that have been working on this topic prior to this thesis. There are some similarities between this thesis and other researchers' works that will be mentioned later in this section. Two of the works will be discussed here that use deep learning algorithms but with different methodologies and deep learning architectures.

One of the works was done by Plamen G. Krastev (2020) regarding the real-time detection of gravitational waves from binary neutron stars using artificial neural networks [13]. In this work, the time series is distinguished between three classes, binary neutron star (BNS), and binary black hole (BBH) merger signals in additive Gaussian noise (signals plus noise), and Gaussian noise only. The neural network used to perform the classification is a convolutional neural network (CNN) that has 4 convo-

lutional and 4 pooling layers, followed by 2 dense fully connected layers. An initial learning rate of 0.001 is used and a batch size of 1000 is chosen with the number of training epochs limited to 100, or until the error on the validation data set stops decreasing. Sparse categorical cross-entropy loss is selected to be the cost function. Figure 2.9 shows the results obtained from this work [13].

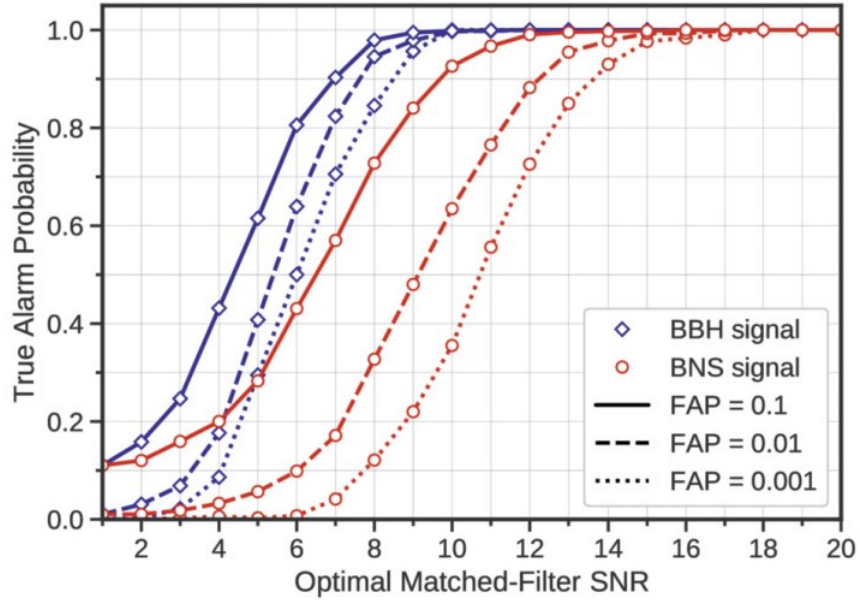


Figure 2.9: Sensitivity curves illustrating the ability of the neural network to identify BNS and BBH GW signals. The true alarm probability (sensitivity) is plotted as a function of the optimal signal-to-noise ratio (SNR) for false alarm probabilities 10^{-1} , 10^{-2} , and 10^{-3} [13].

Furthermore, another similar work was done by Wei Wei (2021) regarding the deep learning for gravitational wave forecasting of neutron star mergers. The ResNet-50 model is pre-trained with ImageNet and its last fully connected layer only outputs a number in the range $[0, 1]$. 60s-long signals with component masses $m_{\{1,2\}} \in [1M_{\odot}, 5M_{\odot}]$ are injected in advanced LIGO data. The evolution of these signals as they sweep through a gravitational wave frequency range between $[20 \text{ Hz}, 150 \text{ Hz}]$

are considered to focus on early detection. These datasets are used to produce a spectrogram dataset to train ResNet-50 using a batch size of 256, and a learning rate of 10^{-4} . In order to improve the robustness of the trained ResNet-50, 50% of the input spectrogram images contain no signals, 25% have simulated signals only in one of the Livingston and Hanford observatories, and the remaining 25% have signals in both Livingston and Hanford strain data. The optimizer used is the ADAM optimizer.

Figure 2.10 shows the results obtained from this work [14].

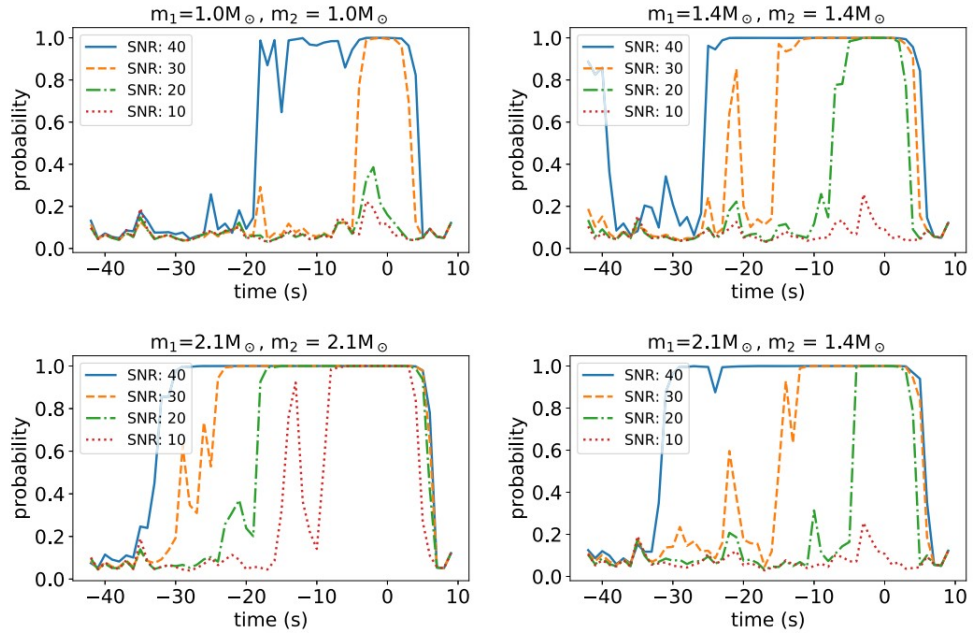


Figure 2.10: Deep learning forecasting for binary neutron stars in advanced LIGO data. An astrophysically motivated sample of binary systems and signal-to-noise ratios show that deep learning identifies signals in real data up to 30 s before merger [14].

The difference between other works and the work in this dissertation is that other researchers do not compare different deep learning models and pick a champion model for classifying the gravitational wave signals. It is important to pick the most suitable deep learning model to classify the gravitational wave signals as misclassification will produce inaccurate results which render one's work invalid. Hence, in this dissertation,

a confusion matrix will be produced for each deep learning model and it is used to calculate the accuracy of each deep learning model. This is to ensure that the results produced will be valid and convincing.

CHAPTER 3

METHODOLOGY

In this section, we will briefly introduce the data description, constant-Q transform (CQT) which is used in the image pre-processing pipeline and transfer learning process. Furthermore, the mechanism of EfficientNet-B0, DenseNet-201 and ResNet-101 which are new possible methods to replace matched filtering technique.

3.1 DATA DESCRIPTION

The dataset employed in this work is provided by G2Net, which is a network of Gravitational Wave, Geophysics and Machine Learning and is established by the European Cooperation in Science and Technology [31] [2]. The G2Net dataset consists of simulated gravitational wave time-series data, each data sample contains 3 time series from 3 different gravitational wave interferometers: LIGO Hanford, LIGO Livingston, and Virgo [2]. Each time series spans 2 seconds and is sampled at 2048 Hz.

The training set file contains 560000 data samples whereas the test set file contains 226000 test samples [31]. However, due to limited computational power, only the first 156250 data samples are chosen from the training set file to be used in the training, validation and testing processes. The distribution of data of the absence and presence of gravitational wave signal is quite even, 78325 and 77925 data samples respectively. In total, there will be 100000 data samples used for training, 25000 data samples for validation and 31250 data samples for testing. Figures 3.1 shows the histogram of the

training set file and the extracted training set file.

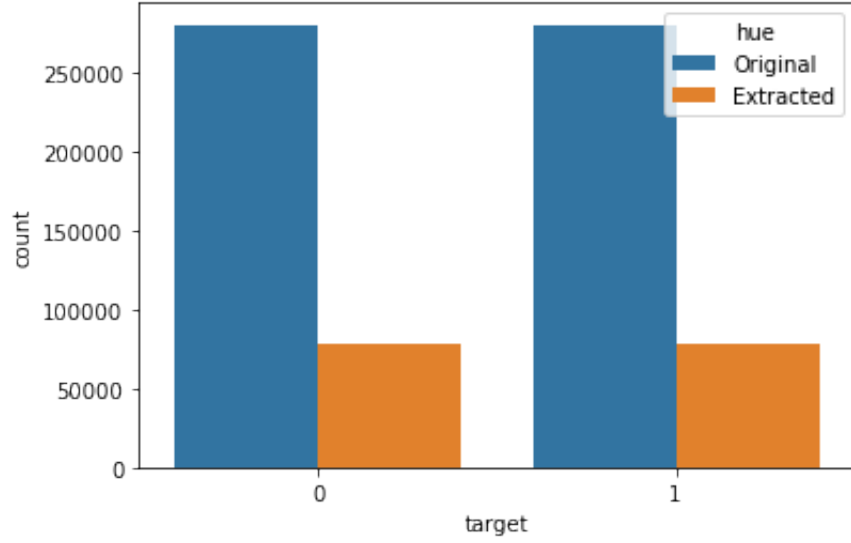


Figure 3.1: Histogram of the original and extracted training set file.

3.2 CONSTANT-Q TRANSFORM

The short-time Fourier transform (STFT) performs a Fourier transform on a short segment which is extracted from a longer data record upon its multiplication with a suitable window function. In order to analyse the local frequency content of the longer data record as a function of time, a sliding window is applied repetitively. The Q factor is defined as the ratio between the center frequency, f_k and the bandwidth, δf :

$$Q = \frac{f_k}{\delta f}. \quad (3.1)$$

Hence, the bandwidth of each filter is constant and the Q factor increases when moving from low to high frequencies since the absolute bandwidth f is identical for all filters. Thus, STFT may not be universally ideal for the time-frequency analysis [15].

Constant-Q transform (CQT) was first introduced in Youngberg and Boll (1978) with an alternative algorithm being proposed by Mont-Reynaud (1986) [42]. CQT was then refined some years later by Brown (1991) [15]. It is a technique that can be used in time-frequency analysis by first transforming a time-domain signal into a time-frequency domain so that the Q-factors are all equal and the center frequencies of the frequency bins are geometrically spaced [43]. Thus, improving the frequency resolution for low frequencies and the time resolution for high frequencies. Figure 3.3 and Figure 3.2 show the comparison of the time-frequency resolution and spectrograms of STFT and CQT [15].

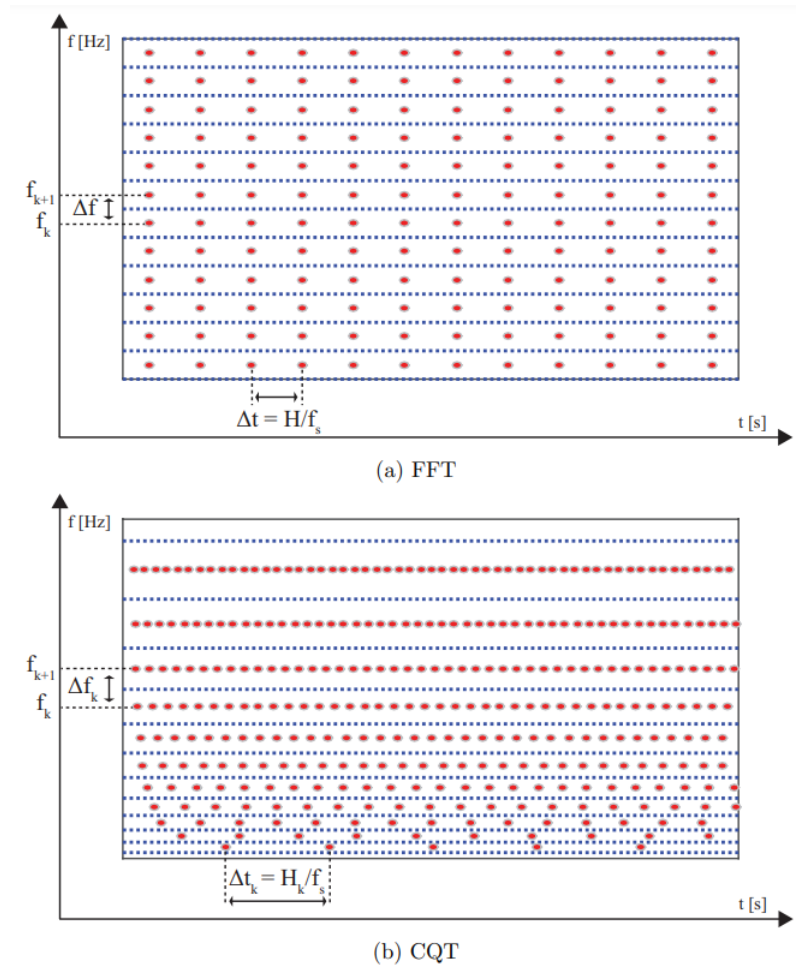


Figure 3.2: Time-frequency resolution of STFT and CQT [15].

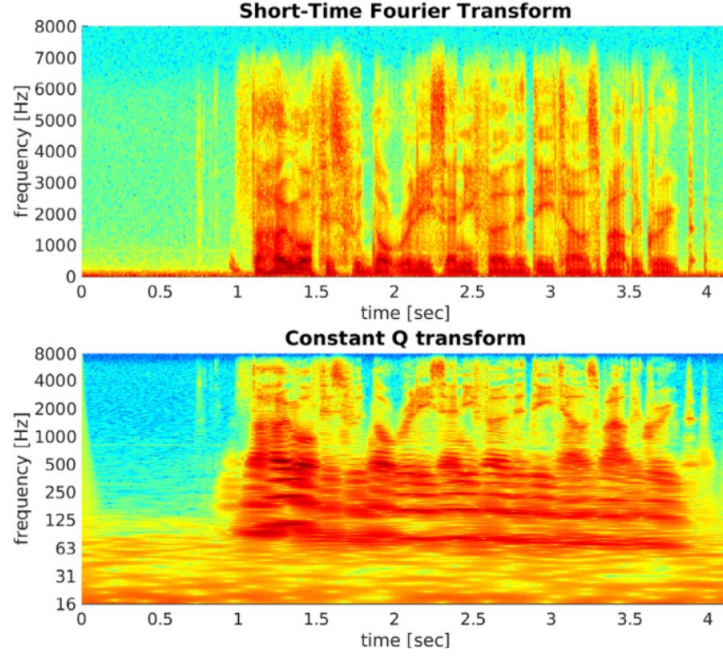


Figure 3.3: Spectrograms of STFT and CQT [15].

The CQT is essentially similar to a wavelet transform with relatively high Q-factors, equivalent to 12-96 bins per octave [15] [43]. An octave is the distance between one note and another note that's double its frequency [44].

The CQT transform $X^{CQ}(k, n)$ of a discrete time-domain signal $x(n)$ is defined by

$$X^{CQ}(k, n) = \sum_{j=n-\lfloor \frac{N_k}{2} \rfloor}^{n+\lfloor \frac{N_k}{2} \rfloor} x(j) a_k^*(j - n + \lfloor \frac{N_k}{2} \rfloor), \quad (3.2)$$

where $a_k^*(n)$ is the complex conjugate of $a_k(n)$, $k = 1, 2, \dots, K$ is the frequency bin index and $\lfloor \cdot \rfloor$ is the floor function [15] [43]. The time-frequency atom, a_k of bin k can be defined by

$$a_k(n) = \frac{1}{N_k} w\left(\frac{n}{N_k}\right) \exp(-2i\pi n \frac{f_k}{f_{sr}}), \quad (3.3)$$

where f_{sr} is the sampling rate, f_k is the center frequency of the lowest frequency bin,

N_k is the window length, and $w(t)$ is a continuous window function. The window function is zero outside the range $t \in [0, 1]$. The window length $N_k \in \mathbb{R}$ are real-valued and inversely proportional to f_k , so the Q-factor is same for all bins k [15] [43].

The center frequencies, f_k can be defined by

$$f_k = 2^{\frac{k-1}{B}} f_1, \quad (3.4)$$

where B is the number of bins per octave and f_1 is the center frequency of the lowest frequency bin. B determines the the time-frequency resolution trade-off of the CQT when using it. The Q-factor of bin k is given by

$$Q_k = \frac{f_k}{\Delta f_k} = \frac{N_k f_k}{\Delta w f_s}, \quad (3.5)$$

where Δw is the -3 dB bandwidth of the mainlobe of the spectrum of the window function $w(t)$, and f_k is the -3 dB bandwidth of the frequency response of the atom $a_k(n)$. By definition, the Q-factors, Q_k are all the same, hence, it can be denoted as just Q [15] [43].

Q should be made as large as possible, in contrast to the bandwidth Δf_k of each bin, which should be made as small as possible. Thus, introducing minimal frequency smearing. The value of Q-factors cannot be arbitrarily high because it will cause some portions of the spectrum between the bins to not be analyzed. The value of Q that introduces minimal frequency smearing but still allows signal reconstruction is

$$Q = \frac{q}{\Delta w (2^{\frac{1}{B}} - 1)}, \quad (3.6)$$

where $q \in (0, 1]$ is the scaling factor, and typically $q = 1$. When $q < 1$, it can be used to improve the time resolution at the cost of degrading the frequency resolution [15] [43].

The window length N_k can be determined by substituting Equations (3.5) and (3.4), N_k is given by

$$N_k = \frac{q f_s}{f_k(2^{\frac{1}{B}} - 1)}, \quad (3.7)$$

where the dependency on Δw has clearly disappeared. In particular, calculating $X^{CQ}(k, n)$ at all positions n of the input signal is not computationally reasonable. Hence, the successive atoms can be placed H_k samples apart, known as hop size. Values $0 < H_k \lesssim \frac{1}{2}N_k$ are useful for analyzing all parts of the signal properly and to achieve reasonable signal reconstruction [2] [15] [43].

In this work, the CQT1992V2 function in the Python package nnAudio [45] is implemented for the constant-Q transform. The parameters are shown in Table 3.1 and Figure 3.4 shows the representation results after applying constant-Q transform.

Table 3.1: Parameters of CQT1992V2 function in this work[2]

Parameter	Symbol	Meaning	Value
sr	f_{sr}	The sampling rate of the input	2048
hop_length	H_k	The hop size	32
fmin	f_1	The center frequency of the lowest frequency bin	20
fmax	-	The frequency of the highest frequency bin, which can be used to determine the number of bins K	1024

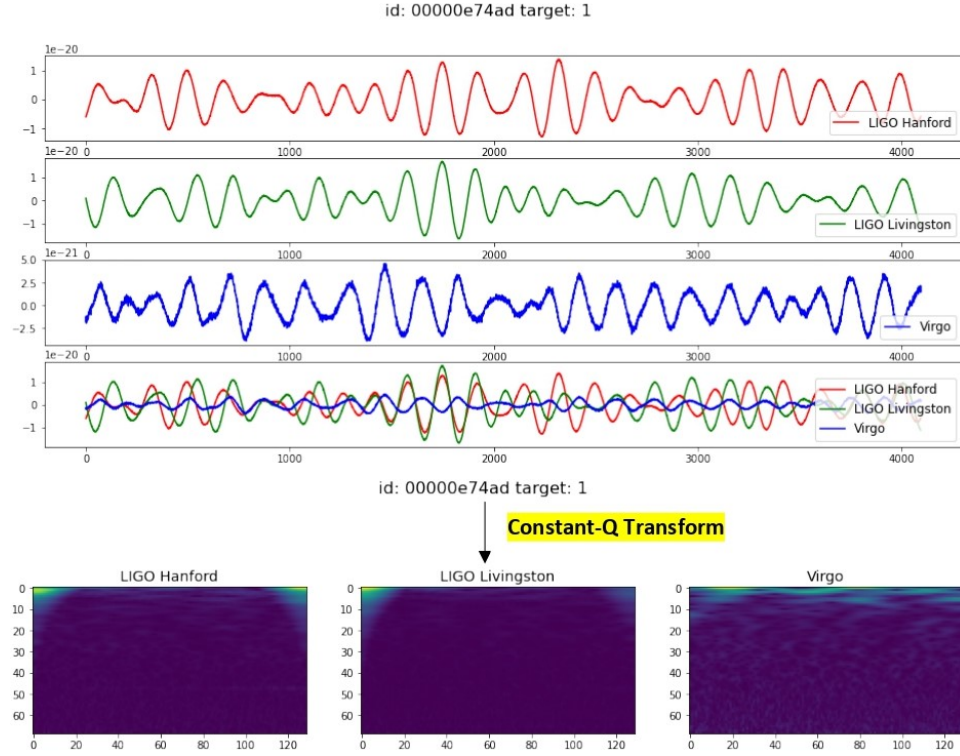


Figure 3.4: Representative data sample in the G2Net dataset and the spectrograms obtained by applying the constant-Q transform.

3.3 TRANSFER LEARNING

CNNs have recently achieved state-of-the-art performance in vision applications, however, there are two major shortcomings that limit their deployments in real life. These shortcomings are that CNNs require large amount of data and high computation power when training from scratch to achieve high accuracy. Hence, transfer learning is designed to overcome these shortcomings by transferring knowledge learned from models pretrained on large datasets which offers the chance for CNNs to learn with limited data samples [46].

Transfer learning techniques have since been applied successfully in many real-world applications. Transfer learning techniques can be used to learn text data across

domains, solving natural language processing and sentiment classification problems [47]. Furthermore, the applications of transfer learning in the field of computer vision include image classification and video concept classification [48]. Moreover, transfer learning techniques are also proposed to extract knowledge from WiFi localization models across mobile devices, time periods, and space in order to benefit WiFi localization tasks in other settings [47].

However, the pre-trained weights cannot be directly used on our gravitational wave dataset due to the difference in the domain of the dataset images and expect high accuracy. Hence, a fine-tuning process is performed to tweak the parameters to adapt to the new domain of images. There are different ways to do fine-tuning which include freezing some weights of some layers and fine-tuning with regularization methods [46].

3.4 EFFICIENTNET

When the width, depth and resolution dimensions of CNNs are scaled up, the model's performance and its accuracy will increase. However, it requires more computational resources to perform such a task. Hence, a new baseline network is developed to obtain a family of models called *EfficientNets* to scale the dimensions uniformly, the method is called compound scaling method [3].

3.4.1 Architecture

Convolutional Neural Network layers are said to be stacked or composed of various convolutional layers: $\mathcal{N} = \mathcal{F}_k \odot \dots \odot \mathcal{F}_2 \odot \mathcal{F}_1(X_1) = \bigodot_{j=1 \dots k} \mathcal{F}_j(X_1)$. The layers are said to be partitioned into multiple stages and all layers in each stage have the same

architecture. Therefore, the CNN can be defined as

$$\mathcal{N} = \bigodot_{i=1\dots s} \mathcal{F}_i^{L_i}(X_{\langle H_i, W_i, C_i \rangle}), \quad (3.8)$$

where \mathcal{F}_i is the convolution operation for the i -th stage, L_i is the number of times \mathcal{F}_i is repeated in stage i , \bigodot denotes the element-wise multiplication, and $\langle H_i, W_i, C_i \rangle$ denotes the shape of input tensor X of layer i [3].

In model scaling, the network length (L_i), width (C_i), and resolution (H_i, W_i) are being expanded without changing \mathcal{F}_i predefined in the baseline network. Model scaling can simplify the design problem by fixing \mathcal{F}_i but there still remains a large design space to explore different L_i, C_i, H_i, W_i for each layer. Hence, all layers must be scaled uniformly with constant ratio in order to further reduce the design space. The target is to maximize the model's accuracy for any given resource constraints, which can be formulated as an optimization problem:

$$\mathcal{N}(d, w, r) = \bigodot_{i=1\dots s} \hat{\mathcal{F}}_i^{d \cdot L_i}(X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, r \cdot \hat{C}_i \rangle}), \quad (3.9)$$

where w, d, r are coefficients for scaling network width, depth, and resolution; $\hat{F}_i, \hat{H}_i, \hat{W}_i, \hat{C}_i$ are predefined parameters in baseline network [3].

However, scaling up any dimension of network width, depth, or resolution improves accuracy but it gains diminishes for bigger models. Figure 3.5 shows the network width scaling under different network depths and resolutions, each dot in a line denotes a model with different width coefficient (w). If the network width w is scaled without changing depth ($d = 1.0$) and resolution ($r = 1.0$), the accuracy saturates

rapidly whereas with deeper ($d = 2.0$) and higher resolution ($r = 2.0$), width scaling achieves much better accuracy under the same FLOPS cost. The number of FLOPs (floating-point operations) here is the direct measurement of computation, the model will be slower if the FLOPs is high [49]. Hence, in order to achieve better accuracy and efficiency, all dimensions of network width, depth, and resolution must be balanced during CNN scaling [3].

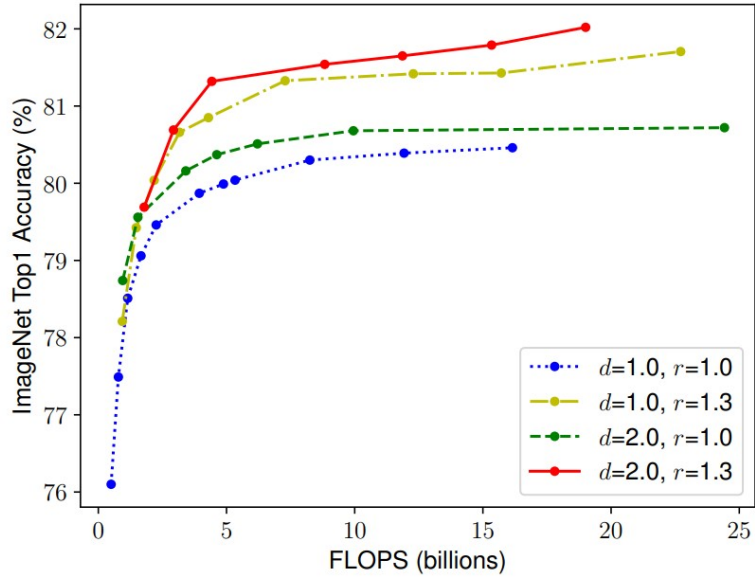


Figure 3.5: Scaling network width for different baseline networks [3].

A new compound scaling method is proposed which uses a compound coefficient ϕ to uniformly scale network width, depth, and resolution in a principled way:

$$\begin{aligned}
&\text{depth : } d = \alpha^\phi, \\
&\text{width : } w = \beta^\phi, \\
&\text{resolution : } r = \gamma^\phi, \\
&\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \\
&\alpha \geq 1, \beta \geq 1, \gamma \geq 1,
\end{aligned} \tag{3.10}$$

where α, β, γ are constants that can be determined by a small grid search. ϕ is a user-defined coefficient that controls how many resources are available for model scaling whereas α, β, γ determine how to assign extra resources to network width, depth, and resolution respectively. The FLOPS of a convolution operation is directly proportional to d, w^2, r^2 , doubling the network depth will double FLOPS. However, doubling the network width or resolution will quadruple FLOPS. Hence, using Equation (3.9) to scale the network will increase the total FLOPS by $(\alpha \cdot \beta^2 \cdot \gamma^2)^\phi$. The constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ is applied such that for any new ϕ , the total FLOPS will approximately increase by 2^ϕ [3].

A new and simple mobile-size baseline architecture called EfficientNet-B0 is developed. Table 3.2 below shows the architecture of EfficientNet-B0 [3].

Table 3.2: EfficientNet-B0 baseline network [3].

Stage i	Operator \hat{F}_i	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Its main building block is mobile inverted bottleneck MBConv to which squeeze-and-excitation optimization is added. Starting from the EfficientNet-B0 baseline network, compound scaling method is applied to scale it up in two steps:

- **STEP 1:** first setting $\phi = 1$ and determine α, β, γ by doing a small grid search. The best values for EfficientNet-B0 are $\alpha = 1.2, \beta = 1.1, \gamma = 1.5$, under the constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$.
- **STEP 2:** α, β, γ are fixed as constants then the baseline network is scaled up with different values of ϕ to obtain EfficientNet-B1 to B7.

Table 3.3 shows the EfficientNet performance results on ImageNet [50]. The baseline network, EfficientNet-B0 is scaled using different compound coefficient ϕ to create other EfficientNet models. Top-1 accuracy measures the proportion of examples for which the predicted label matches the single target label whereas top-5 accuracy considers a classification correct if any of the five predictions matches the target label [51].

Table 3.3: EfficientNet Performance Results on ImageNet [3].

Model	Top-1 Acc.	Top-5 Acc.	#Params	#FLOPS
EfficientNet-B0	77.1%	93.3%	5.3M	0.39B
EfficientNet-B1	79.1%	94.4%	7.8M	0.70B
EfficientNet-B2	80.1%	94.9%	9.2M	1.0B
EfficientNet-B3	81.6%	95.7%	12M	1.8B
EfficientNet-B4	82.9%	96.4%	19M	4.2B
EfficientNet-B5	83.6%	96.7%	30M	9.9B
EfficientNet-B6	84.0%	96.8%	43M	19B
EfficientNet-B7	84.3%	97.0%	66M	37B

In this work, the EfficientNet-B0 model will be implemented due to limited computational resources. To ensure high efficiency and accuracy, the original fully connected layers are removed and a few layers such as dense and dropout layers are added to EfficientNet-B0 as new fully connected layers.

3.4.2 Modified Architecture

The EfficientNet-B0 architecture is first loaded, then its fully connected layers are omitted by setting *include_top = False*. A new fully connected layer head is added on top of the base network of EfficientNet-B0 which consists of a dropout layer in order to achieve better accuracy and efficiency. Resizing, conv2d and batch normalization layers are added before the EfficientNet-B0 architecture. Table 3.4 shows the modified architecture of EfficientNet-B0.

Table 3.4: Modified architecture of EfficientNet-B0. The highlighted row is the original architecture without an FC layer.

Layer (Type)	Output Shape	Param #
resizing (Resizing)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 224, 224, 3)	84
batch_normalization (BatchNormalization)	(None, 224, 224, 3)	12
efficientnetb0 (Functional)	(None, 1280)	4049571
Dropout (0.2)	(None, 1280)	0
dense (Dense)	(None, 1)	1281

Dense layer connects every neuron in the defined layer to every neuron in the previous layer. The dropout layer drops out a few neurons or sets them to 0 and reduces computation in the training process. For instance, *Dropout* (0.2) layer will drop out 20% of the neurons [40]. The Batch Normalization layer¹ applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.

Conv2D layer² is a 2D Convolution layer that creates a convolution kernel that convolves with the layer input to produce a tensor of outputs. In this work, *padding* = “same” which the output has the same size as the input. The kernel regularizer is set to be *L2* class³ that applies a L2 regularization penalty. The Resizing layer⁴ resizes the image input to a target height and width. In this work, the original image size (69, 129, 3) is resized to (224, 224, 3), which is the image input size of EfficientNet-B0. The dense layer in the fully connected layer is set as *activation* = “sigmoid” to renders the output between 0 and 1 [40].

¹https://keras.io/api/layers/normalization_layers/batch_normalization/

²https://keras.io/api/layers/convolution_layers/convolution2d/

³<https://keras.io/api/layers/regularizers/>

⁴https://keras.io/api/layers/preprocessing_layers/image_preprocessing/resizing/

3.4.3 Fine-tuning

All the layers for EfficientNet-B0 recieved fine-tuning such as setting *pooling* = “*avg*” and the pre-trained weights all came from ImageNet, which contains images from 1000 different classes of day-to-day objects [50] [52]. Besides that, the L2 regularization is used in the conv2d layer and some dense layers from the new fully connected layers. The optimizer function used here is Adam with a learning rate of 0.001.

3.5 RESNET

Deep neural networks are more difficult to train, so a residual learning framework is introduced to ease the training that are substantially deeper. The layers are formulated as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. This shows that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. Hence, a family of models called *ResNets* is created [4].

3.5.1 Architecture

The training accuracy of a deep neural network can be improved by stacking more layers, creating deeper networks. However, a degradation problem arises when the deep networks start to converge. In which, with the network depth increasing, the accuracy gets saturated and then degrades rapidly. Overfitting does not cause such degradation, and adding more layers leads to higher training error. As such, the solution is by adding construction to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model, in which these layers are fitted a

residual mapping. Thus, the model should not produce high training error as compared to its shallower counterparts [4].

Hence, the degradation problem is addressed by introducing a deep residual learning framework. Lets consider $\mathcal{F}(x)$ to be the residual function and $\mathcal{H}(x)$ as an underlying mapping to be fitted by a few stacked layers with x denoting the inputs to the first few layers.

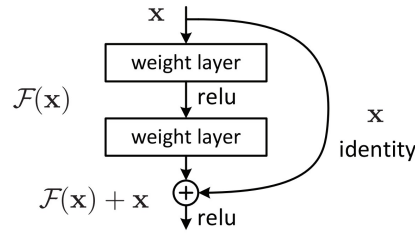


Figure 3.6: Residual learning: a building block [4].

These layers are said to approximate a residual function $\mathcal{F}(x) := \mathcal{H}(x) - x$, so the original function becomes $\mathcal{H}(x) := \mathcal{F}(x) + x$ as shown in Figure 3.6. As mentioned previously, deeper models should have training error no greater than its shallower counterpart if the added layers can be constructed as identity mappings, and if identity mappings are optimal, the solvers may simply drive the weights of the multiple non-linear layers toward zero to approach identity mappings. However, identity mappings are unlikely to be optimal, but the reformulation may help to precondition the problem. It should be easier for the solver to find the perturbations with reference to an identity mapping, than to learn the function as a new one if the optimal function is closer to an identity mapping than to a zero mapping [4].

A residual function can be defined as:

$$y = \mathcal{F}(x, \{W_i\}) + x, \quad (3.11)$$

where $\mathcal{F}(x, \{W_i\})$ is the residual mapping to be learned, x and y are the input of the residual and the output function respectively, with \mathcal{F} and x having equal dimensions. If the dimensions of \mathcal{F} and x are not equal in Equation (3.10), a linear projection W_s can be performed by the shortcut connections to match the dimensions:

$$y = \mathcal{F}(x, \{W_i\}) + W_s x, \quad (3.12)$$

where W_s is only used when matching dimensions. Although the above notations are about fully-connected layers, they are applicable to convolutional layers. The function $\mathcal{F}(x, \{W_i\})$ can represent multiple convolutional layers and the element-wise addition is performed on two feature maps, channel by channel [4]. Table 3.5 below shows the architectures of different ResNet models.

Table 3.5: Architectures of different ResNet models [4].

Layer Name	Output Size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3, max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 2$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	Average pool, 1000-d fc, softmax				
FLOPS		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 3.6 shows the error rates of single-model results on the ImageNet validation set of ResNet models.

Table 3.6: Error rates of single-model results on the ImageNet of ResNet models [4].

Model	Top-1 err.	Top-5 err.
ResNet-50	20.74%	5.25%
ResNet-101	19.87%	4.60%
ResNet-152	19.38%	4.49%

In this work, the ResNet-101 model will be implemented due to limited computational resources. To ensure high efficiency and accuracy, the original fully connected layers are removed and few more layers such as dense and dropout layers have been added to ResNet-101 as new fully connected layers.

3.5.2 Modified Architecture

The ResNet-101 architecture is first loaded, then its fully connected layers are omitted by setting *include_top = False*. A new fully connected layer head is added on top of the base network of ResNet-101 which consists of a few dense and dropout layers in order to achieve better accuracy and efficiency. Resizing, conv2d and batch normalization layers are added before the ResNet-101 architecture. Table 3.7 shows the modified architecture of ResNet-101.

Table 3.7: Modified architecture of ResNet-101. The highlighted row is the original architecture without an FC layer.

Layer (Type)	Output Shape	Param #
resizing (Resizing)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 224, 224, 3)	84
batch_normalization (BatchNormalization)	(None, 224, 224, 3)	12
resnet101 (Functional)	(None, 2048)	42658176
dense (Dense)	(None, 1024)	2098176
dense_1 (Dense)	(None, 512)	524800
dense_2 (Dense)	(None, 256)	131328
Dropout (0.3)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
Dropout (0.2)	(None, 128)	0
dense_4 (Dense)	(None, 1)	129

The dense layer connects every neuron in the defined layer to every neuron in the previous layer. The dropout layer drops out a few neurons or sets them to 0 and reduces computation in the training process [40]. The Batch Normalization layer applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.

The modified ResNet-101 architecture has a 2D Convolution layer as well. The kernel regularizer is set to be *L2* class that applies a L2 regularization penalty. The resizing layer resizes the original image size (69, 129, 3) to (224, 224, 3), which the image input size of ResNet-101. The activation function function of each layer is set *activation* = “*relu*”, aside from the last dense layer in the fully connected layers which is set *activation* = “*sigmoid*” [40].

3.5.3 Fine-tuning

All the layers for ResNet-101 recieved fine-tuning such as setting *pooling* = “*max*” and the pre-trained weights all came from ImageNet [52]. Besides that, the L2 regularization is used in the conv2d layer and some dense layers from the new fully connected layers. The optimizer function used here is Stochastic Gradient Descent (SGD) with a learning rate of 0.001 and a momentum of 0.9.

3.6 DENSENET

The convolutional networks that contain shorter connections between layers close to the input and those close to the output can be substantially deeper, more accurate, and efficient to train. Hence, Dense Convolutional Network (DenseNet) is created to connect each layer to every other layer in a feed-forward fashion. DenseNets alleviate the vanishing-gradient problem, strengthens feature propagation, encourages feature reuse, and substantially reduces the number of parameters [5].

3.6.1 Architecture

A new research problem emerges when CNNs get deeper: information about the input or gradient can vanish by the time it reaches the end, or beginning of the network when passed through many layers. A solution is proposed to solve this issue by connecting all layers with matching feature-map sizes directly with each other to ensure maximum information flow between layers in the network. Each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers to preserve the feed-forward nature. DenseNets require fewer parameters as compared to

traditional convolutional networks and improve the flow of information and gradients throughout the network, which makes them easy to train [5]. Figure 3.7 shows the layout schematically.

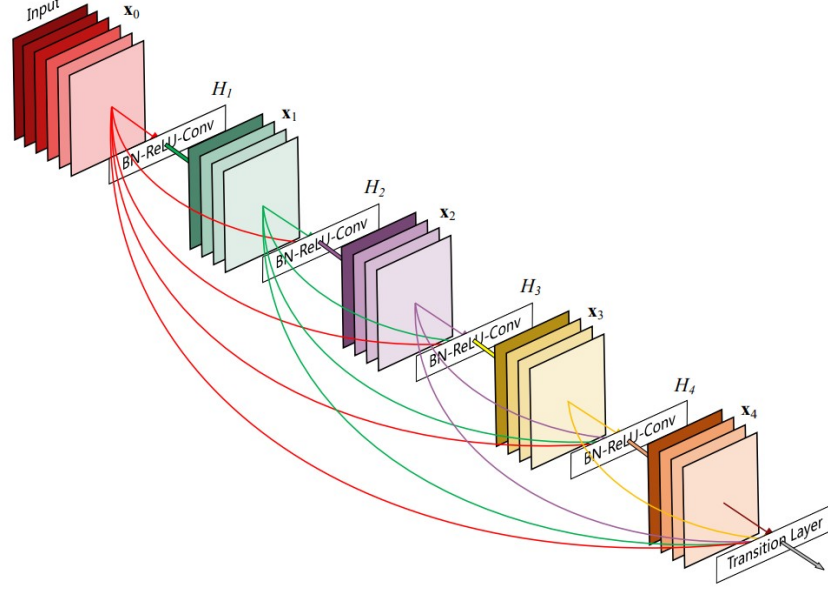


Figure 3.7: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input [5].

Consider a convolutional network comprising of L layers, each of which implements a non-linear transformation $H_\ell(\cdot)$, where ℓ indexes the layer. $H_\ell(\cdot)$ is the composite function of operations which includes rectified linear units (ReLU), batch normalization (BN), convolution (Conv), or pooling. A single image x_0 passes through the network and the output of the ℓ^{th} layer is denoted as x_ℓ . DenseNets implement direct connections from any layer to all subsequent layers, the ℓ^{th} layer receives the feature-maps of all preceding layers, $x_0, \dots, x_{\ell-1}$, as inputs:

$$x_\ell = H_\ell([x_0, x_1, \dots, x_{\ell-1}]), \quad (3.13)$$

where $[x_0, x_1, \dots, x_{\ell-1}]$ refers to the concatenation of the feature-maps produced in layers $0, \dots, \ell - 1$. The multiple inputs of $H_\ell(\cdot)$ in Equation (3.12) can be concatenated into a single tensor [5].

The concatenation operation cannot be done if the size of feature-maps changes. Therefore, the network is divided into multiple densely connected dense blocks as shown in Figure 3.8 below.

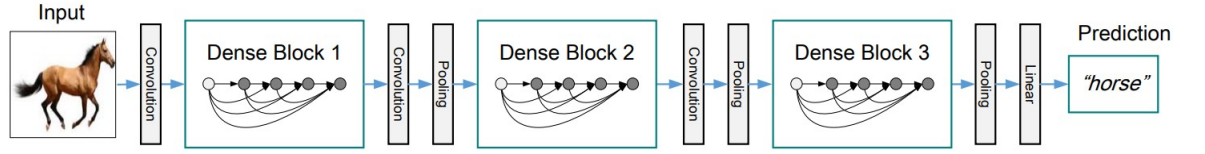


Figure 3.8: A deep DenseNet with three dense blocks [5].

The layers between the blocks are transition layers that change the size of feature-maps, which do convolution and pooling. The transition layers consist of a batch normalization layer and an 1×1 convolutional layer and a 2×2 average pooling layer. The number of channels will increase every layer, as shown below assuming each function H_ℓ produces k feature-maps

$$k_\ell = k_0 + k \times (\ell - 1), \quad (3.14)$$

where k_ℓ is the number of channels in the ℓ^{th} layer, k_0 is the number of channels in the input layer and k is the growth rate. The growth rate regulates how much new information is added into the network in each layer [5]. Table 3.8 shows the architectures of different DenseNet models.

Table 3.8: Architectures of different DenseNet models [5].

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56 28×28	1×1 conv 2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28 14×14	1×1 conv 2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14 7×7	1×1 conv 2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool 1000D fully-connected, softmax			

Table 3.9 shows the error rates on the ImageNet validation set with single-crop testing of DenseNet models.

Table 3.9: Error rates on the ImageNet validation set with single-crop testing of DenseNet models [5].

Model	top-1	top-5
DenseNet-121	25.02%	7.71%
DenseNet-169	23.80%	6.85%
DenseNet-201	22.58%	6.34%
DenseNet-264	22.15%	6.12%

In this work, the DenseNet-121 model will be implemented due to limited computational resources. To ensure high efficiency and accuracy, the original fully connected layers are removed and few more layers such as dense and dropout layers have been added to DenseNet-121 as new fully connected layers.

3.6.2 Modified Architecture

The DenseNet-121 architecture is first loaded, then its fully connected layers are omitted by setting *include_top = False*. A new fully connected layer head is added on top of the base network of DenseNet-121 which consists of a few dense and dropout layers in order to achieve better accuracy and efficiency. Resizing, conv2d and batch normalization layers are added before the DenseNet-121 architecture. Table 3.10 shows the modified architecture of DenseNet-121.

Table 3.10: Modified architecture of DenseNet-121. The highlighted row is the original architecture without an FC layer.

Layer (Type)	Output Shape	Param #
resizing (Resizing)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 224, 224, 3)	84
batch_normalization (BatchNormalization)	(None, 224, 224, 3)	12
densenet121 (Functional)	(None, 2048)	7037504
dense (Dense)	(None, 1024)	1049600
dense_1 (Dense)	(None, 512)	524800
dense_2 (Dense)	(None, 256)	131328
Dropout (0.3)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
Dropout (0.2)	(None, 128)	0
dense_4 (Dense)	(None, 1)	129

Explanations for conv2d, resizing, batch normalization, dense, and dropout layers are explained in previous sections.

3.6.3 Fine-tuning

All the layers for DenseNet-121 recieved fine-tuning such as setting *pooling = "max"* and the pre-trained weights all came from ImageNet [52]. Besides that, the L2 regularization is used in the conv2d layer and some dense layers from the new fully connected layers. The optimizer function used is the same as the one used in ResNet-101.

3.7 LOSS FUNCTION

A loss function measures the loss from the target which helps a network understand whether it is improving its learning process in each iteration. It is a mathematical concept associated with optimization theory. A loss function can be categorized into two groups, classification (discrete) and regression (continuous) [40] [53].

Since this work involves the classification of gravitational wave signals, only losses for categorical outcomes will be discussed in this section. For categorical outcomes, the final prediction will either be binary or multi-class and the two primary types of loss functions for categorical outcomes are shown below

$$Loss = \sum_{i=1}^n [-y_i \ln(p_i) - (1 - y_i) \ln(1 - p_i)] \quad (\text{Binary cross-entropy}), \quad (3.15)$$

$$Loss = - \sum_{i=1}^n y_i \log_2 \hat{y}_i \quad (\text{Categorical cross-entropy}). \quad (3.16)$$

Binary cross-entropy defines the loss when the categorical outcomes is a binary variable whereas categorical cross-entropy defines the loss when the categorical outcomes is multi-class [40] [53].

3.8 PERFORMANCE EVALUATION METRICS

This section describes the evaluation metrics as the measurement tool that measures the performance of classifier, mainly accuracy and confusion matrix [52] [54]. The evaluation metrics are employed to select the optimal solution which can produce a more accurate prediction of future evaluation of a classifier and obtain an optimized classifier [54].

3.8.1 Confusion Matrix

One of the evaluation metrics used in this work to evaluate the classifier is the confusion matrix. It is a correlation matrix between the actual label and the predicted label of the model [52]. Figure 3.9 shows the graphical representation of a confusion matrix for binary classification.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 3.9: Confusion matrix for binary classification [16].

From this confusion matrix, *True Positive (TP)* and *True Negative (TN)* denote the number of positive and negative instances that are correctly classified. Meanwhile, *False Positive (FP)* and *False Negative (FN)* denote the number of misclassified negative and positive instances respectively [52] [54].

3.8.2 Accuracy

Another evaluation metric is the accuracy which refers to a measure of the degree to how accurately it classifies both positives and negatives [12] [55]. In other words, the ratio of correctly predicted image classes to the total number of images in a dataset [52]. The formula of accuracy is shown below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3.17)$$

where *False Positive* (FP) and *False Negative* (FN) denote the number of misclassified negative and positive instances respectively [52] [54].

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, we will dive into the results yielded from the three deep learning models used in this work. There are three results that have been obtained from the deep learning models, the accuracy, confusion matrix, and loss function. The champion model will be chosen from one of the three models with the highest accuracy.

4.1 EFFICIENTNET-B0

EfficientNet uses the compound scaling method to scale the width, depth, and resolution dimensions uniformly. EfficientNet-B0 is the baseline network and the other EfficientNet models are scaled from this baseline network using different compound coefficient ϕ . However, in this work, the variant of EfficientNet used to perform the classification task is EfficientNet-B0.

4.1.1 Confusion Matrix

The confusion matrix of EfficientNet-B0 obtained in this work is shown in Figure 4.1

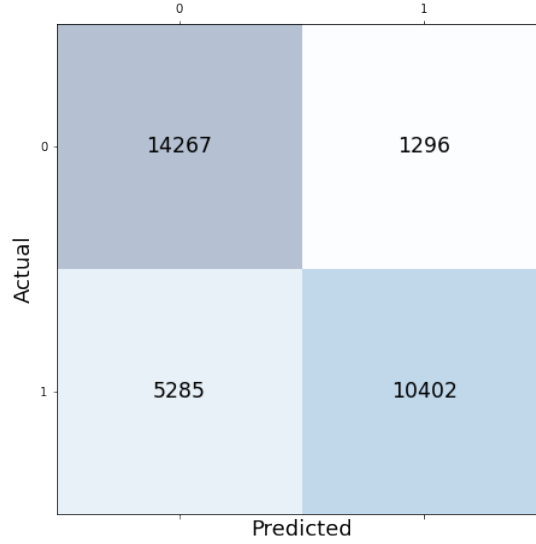


Figure 4.1: Confusion matrix of EfficientNet-B0. The top left and right of the confusion matrix show TN and FP respectively. Whereas the bottom left and right show FN and TP respectively.

Based on the confusion matrix obtained, the number of correctly classified gravitational waves signals and the number of correctly classified images with no signal are 10402 and 14267 respectively. Whereas the number of False Positives (FPs) and False Negatives (FNs) classified are 1296 and 5285 respectively.

4.1.2 Accuracy

The accuracy of EfficientNet-B0 assessed on G2Net dataset is obtained as shown in the equation below

$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\
 &= \frac{10402 + 14267}{10402 + 14267 + 1296 + 5285} \\
 Accuracy &= 0.789408
 \end{aligned}$$

Based on the calculation shown above, the accuracy of EfficientNet-B0 assessed on G2Net dataset obtained is 0.789408 or about 78.9%. Figure 4.2 shows the training and validation accuracy of ResNet-101 obtained in this work.

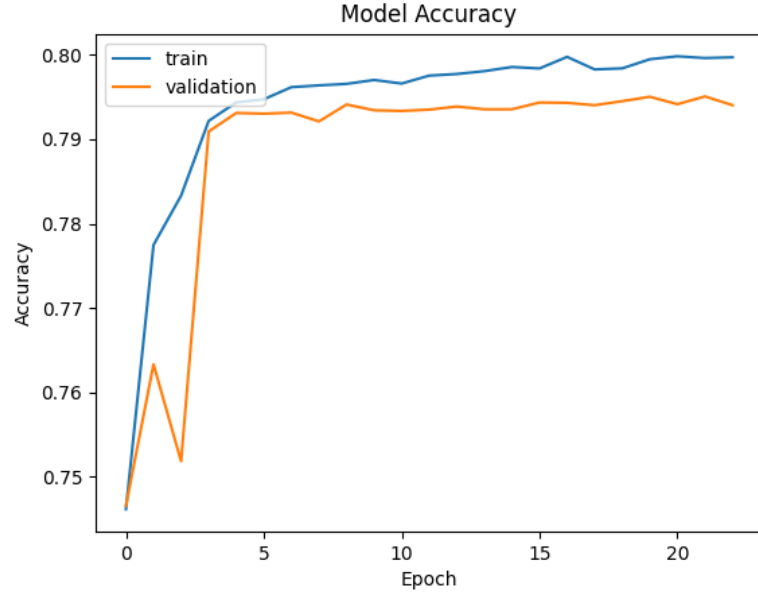


Figure 4.2: Training and validation accuracy of EfficientNet-B0.

Based on the results obtained in this work, the training accuracy increased gradually after each epoch whereas the validation accuracy has a small drop at the third epoch but then having a spike at the fourth epoch. The validation accuracy plateaued after the fourth epoch whereas the training accuracy kept increasing slightly, approaching the accuracy of 0.80.

In this work, the model is set to run for 30 epochs, however, the EarlyStopping¹ function is used to stop the training process when the monitored metric has stopped improving, in this case the monitored metric is 'val_loss', the validation loss and the

¹https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

training process stopped at twenty-third epoch. The ReduceLROnPlateau² function is used to reduce the learning rate when the monitored metric has stopped improving, in this case the monitored metric is 'val_loss', the validation loss as well. The factor by which the learning rate will be reduced and lower bound on the learning rate of ReduceLROnPlateau function is set to be $factor = 0.01$ and $min_lr = 0.00001$ respectively.

4.1.3 Loss Function

Figure 4.3 shows the training and validation losses graph of EfficientNet-B0 obtained in this work

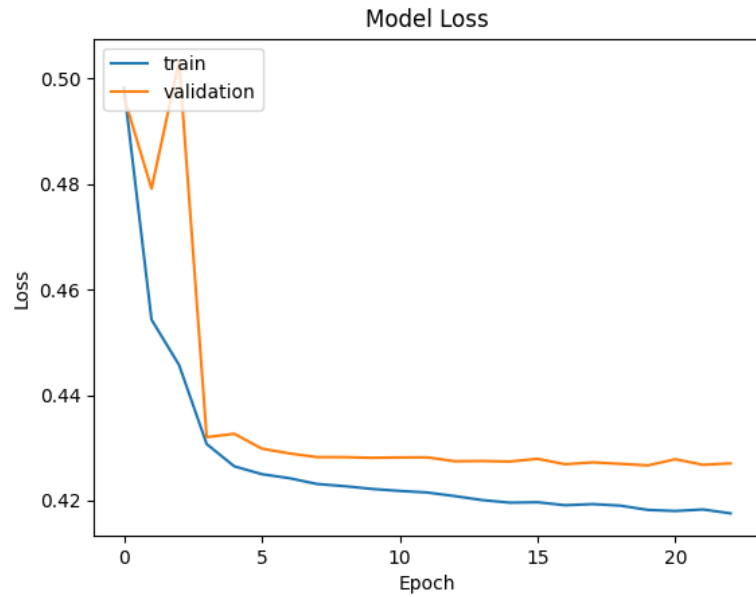


Figure 4.3: Training and validation losses of EfficientNet-B0.

The graph shows that the training loss decreases exponentially whereas the validation loss decreased at first then having a spike at the third epoch which matches the drop in

²https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ReduceLROnPlateau

accuracy as shown in Figure 4.2. Then a big drop at the fourth epoch which matches the raise in accuracy in Figure 4.2 as well. However, this still indicates that the weights are being updated correctly and the loss is still being reduced after each epoch even though there is a drop in accuracy and a raise in loss at third epoch. The validation loss started to plateau after the fourth epoch whereas the training loss kept decreasing slightly which almost plateaued at 0.41.

4.2 RESNET-101

ResNet uses identity mappings to solve the degradation problem which causes the accuracy gets saturated and then degrades rapidly. In this work, the variant of ResNet used to perform the classification task is ResNet-101.

4.2.1 Confusion Matrix

The confusion matrix of ResNet-101 obtained in this work is shown in Figure 4.4

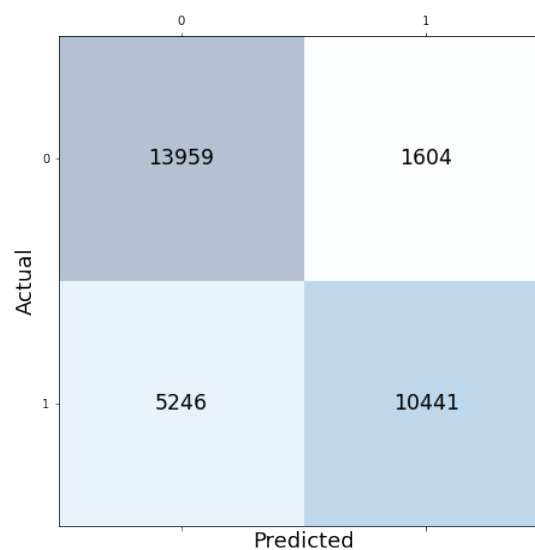


Figure 4.4: Confusion matrix of ResNet-101.

Based on the confusion matrix obtained, the number of correctly classified gravitational waves signals and the number of correctly classified images with no signal are 10441 and 13959 respectively. Whereas the number of False Positives (FPs) and False Negatives (FNs) classified are 1604 and 5246 respectively.

4.2.2 Accuracy

The accuracy of ResNet-101 assessed on G2Net dataset obtained is 0.7808 or about 78.1%. Figure 4.5 shows the training and validation accuracy of ResNet-101 obtained in this work.

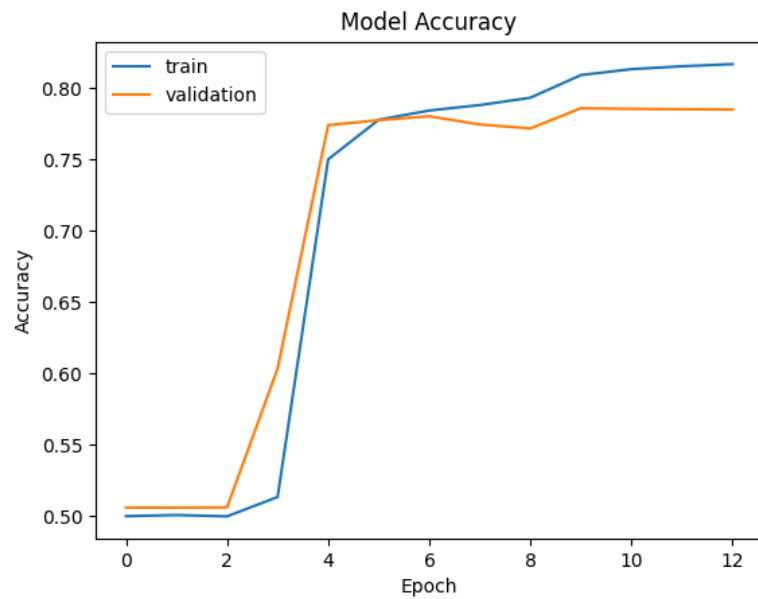


Figure 4.5: Training and validation accuracy of ResNet-101.

Based on the results obtained in this work, both accuracy increased after the second epoch with the validation accuracy being higher than training accuracy. However, after the fifth epoch, the training accuracy kept increasing while the validation accuracy starting to plateau.

In this work, the model is set to run for 30 epochs, however, the EarlyStopping function stopped the training process at thirteenth epoch. The factor by which the learning rate will be reduced and lower bound on the learning rate of ReduceLROnPlateau function is set to be $factor = 0.01$ and $min_lr = 0.00001$ respectively.

4.2.3 Loss Function

Figure 4.6 shows the training and validation losses graph of ResNet-101 obtained in this work

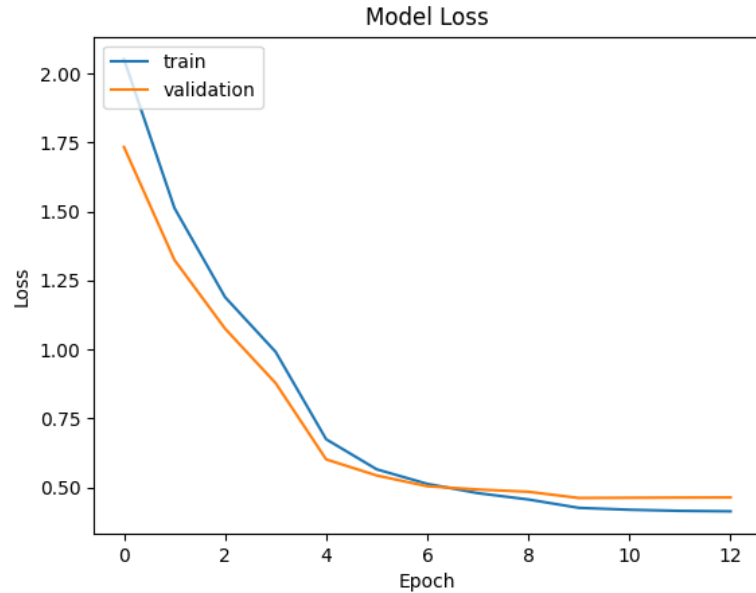


Figure 4.6: Training and validation losses of ResNet-101.

The graph shows that the loss function decreases exponentially which indicates that the weights are being updated correctly and the losses are reduced after each epoch. It started off with the training loss higher than validation loss but at the sixth epoch, it was overtaken by the validation loss, being slightly higher than the training loss. However, both losses started to plateau after the sixth epoch.

4.3 DENSENET-121

DenseNet connects all layers with matching feature-map sizes directly with each other to ensure maximum information flow between layers in the network. In this work, the variant of DenseNet used to perform the classification task is DenseNet-121.

4.3.1 Confusion Matrix

The confusion matrix of DenseNet-121 obtained in this work is shown in Figure 4.7

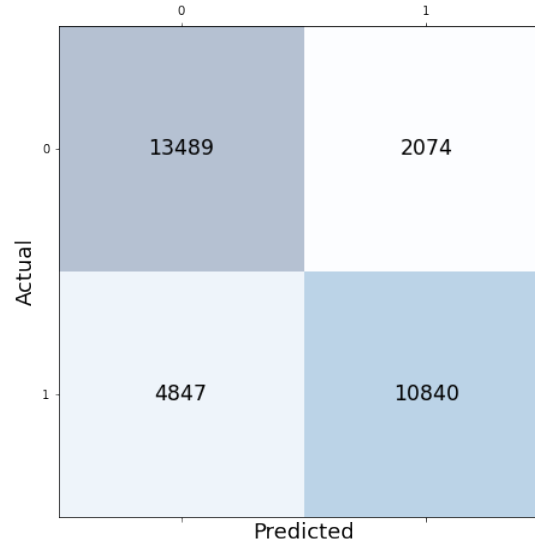


Figure 4.7: Confusion matrix of DenseNet-121.

Based on the confusion matrix obtained, the number of correctly classified gravitational waves signals and the number of correctly classified images with no signal are 10840 and 13489 respectively. Whereas the number of False Positives (FPs) and False Negatives (FNs) classified are 2074 and 4847 respectively.

4.3.2 Accuracy

The accuracy of DenseNet-121 assessed on G2Net dataset obtained is 0.778528 or about 77.9%. Figure 4.8 shows the training and validation accuracy of DenseNet-121 obtained in this work.

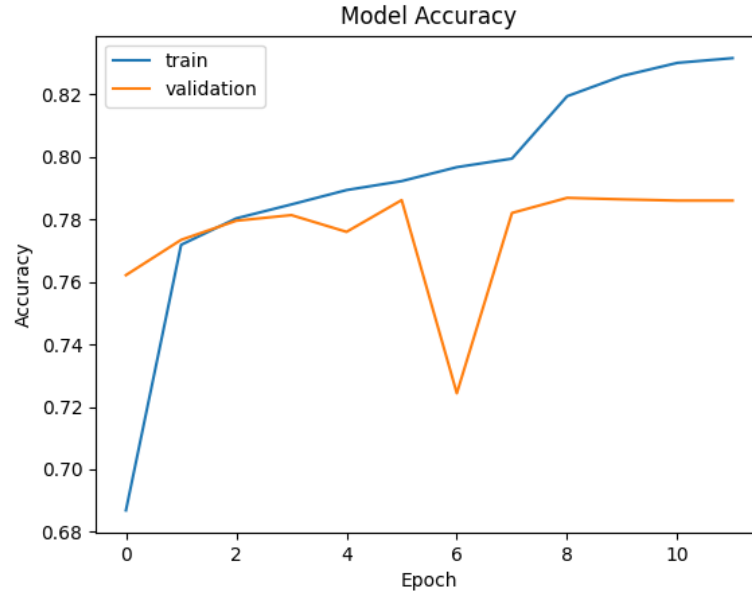


Figure 4.8: Training and validation accuracy of DenseNet-121.

Based on the results obtained in this work, the validation accuracy started off being higher than training accuracy but it was overtaken after the second epoch. The training accuracy kept increasing while the validation accuracy plateaued at around 0.78 accuracy with a huge drop down to 0.72 at the sixth epoch.

In this work, the model is set to run for 30 epochs, however, the EarlyStopping function the EarlyStopping function stopped the training process at twelfth epoch. The EarlyStopping and ReduceLROnPlateau functions' hyperparameters are the same as the ones used in ResNet-101.

4.3.3 Loss Function

Figure 4.9 shows the training and validation losses graph of DenseNet-121 obtained in this work

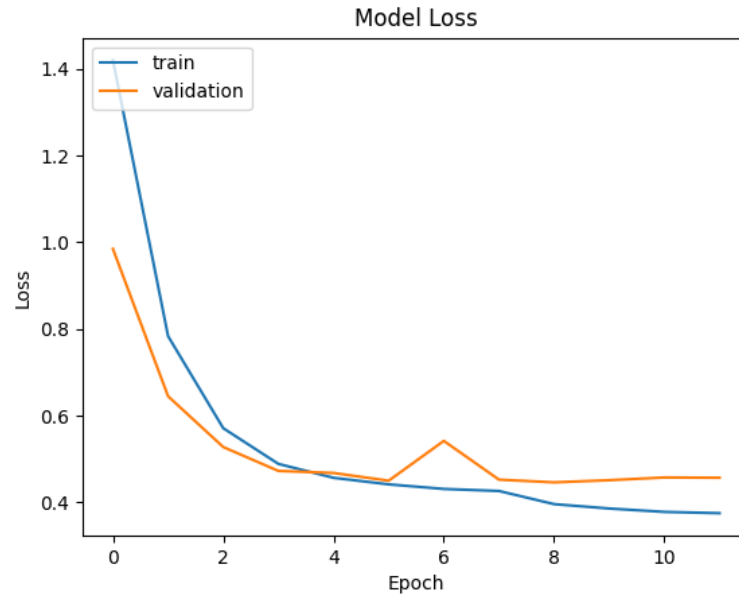


Figure 4.9: Training and validation losses of DenseNet-121.

The graph shows that the loss function decreases exponentially which indicates that the weights are being updated correctly and the losses are reduced after each epoch. It started off with the training loss higher than validation loss but at the fourth epoch, it was overtaken by the validation loss, being higher than the training loss. Both losses started to plateau after the fourth epoch with a spike at the sixth epoch for the validation loss which matches the slight drop in the validation accuracy as shown in Figure 4.8.

4.4 DISCUSSION

Based on all the results obtained in this work, the champion model is EfficientNet-B0 which it outperformed other models due to the fact that it has the highest accuracy, 78.9% with a minimum loss function of 0.4267. Hence, EfficientNet-B0 can be used in the future to perform gravitational wave signals classification task. However, there are more improvements that can be made to further improve the model's accuracy and reduce its loss function. These improvements will be further discussed in the next chapter.

In the early stage of this work, a variant of the EfficientNet family was used without transfer learning, so the model can be trained from scratch using all of the data samples given in the file. The variant used was EfficientNet-B7 along with 560000 data samples being used to train the model. However, such approach was abandoned due to EfficientNet-B7 having more parameters and more data samples were used which increased the computation time, the time taken to complete the training and testing processes is about 7 days. Hence, a new approach like using transfer learning, fewer data samples, and a different variant of EfficientNet to reduce computation time.

According to the accuracy graphs for all three models, EfficientNet-B0 has the most epochs among the graphs, 23 epochs. This is due to the validation loss of EfficientNet-B0 kept changing and improving by the ReduceLROnPlateau function. Hence, it took 23 epochs for the validation loss to finally stopped improving and the training process is thus terminated by the EarlyStopping function. However, this has resulted a higher accuracy for EfficientNet-B0 as compared to other models.

Furthermore, according to the confusion matrix of each model, their number of False Negatives (FNs) has always been higher than the number of False Positives (FPs). This may be due to the fact that some spectrograms do contain the gravitational wave signals, however, the signals may not be as obvious as others. Which in turn, causing the models to wrongly classify these spectrograms as not containing the gravitational wave signals.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this final section, we will conclude our thesis based on the results obtained in the previous chapter and discuss about how deep learning methods can be further improved in future work.

5.1 CONCLUSION

In this thesis, we have studied the different results obtained from implementing different deep learning models to classify simulated gravitational wave signals from noises provided in the G2Net dataset. The performance of each deep learning model is compared to other in order to choose the champion model. The hyperparameters of each deep learning model are tuned to optimize its loss function. The results obtained are summarized in the following points:

- **EfficientNet-B0** performs the best among all the three deep learning models with an accuracy of 78.9% and a minimum loss function of 0.4267.
- **ResNet-101** performs better than DenseNet-121 but slightly worse than EfficientNet-B0 with an accuracy of 78.1% and a minimum loss function of 0.4620.
- **DenseNet-121** performs the worst among all three deep learning models with an accuracy of 77.9% and a minimum loss function of 0.4465.

In conclusion, based on the results obtained in this work, we can clearly conclude

that **EfficientNet-B0** is the champion model with the highest accuracy and the lowest loss function. EfficientNet-B0 is suitable to be used to perform the gravitational wave signals classification task.

5.2 FUTURE WORK

In this work, we have explored various ways in improving the performance of the deep learning models such as using transfer learning and weight regularization. However, we discovered that there are more ways that can still be implemented to further improve the accuracy and loss function of our deep learning models. In the following, we list out the possible ways that we can further improve our deep learning models in future work:

- Having higher computational power to increase the batch size, reduce time cost, use a better variant of the deep learning model and thus, increase the image size.
- Applying K-Fold cross validation to improve our model by resampling different data portions for testing and training of model on different iterations, hence preventing overfitting and rendering the highest accuracy score.
- Using different optimizer functions such as RMSprop, Nadam, Adagrad, and Adadelata to get better accuracy and lower loss function.
- Implement other regularizers such as bias regularizer that applies a penalty on the layer's bias and activity regularizer that applies a penalty on the layer's output.
- Implement kernel initializer such as HeNormal, HeUniform, RandomNormal, and GlorotNormal for the kernel weights matrix.

- Implement different activation functions such as ‘tanh’, ‘selu’, ‘softplus’, and ‘exponential’ other than ‘relu’ and ‘sigmoid’.
- Freeze all layers and only train the top layers then unfreeze all or a fraction of the layers to proceed with the training and validating processes.

By using the ways mentioned above, we could be able to improve the models’ accuracy and reduce their loss functions. Furthermore, we can then replace conventional match-filtering techniques with fully optimized deep learning algorithms. We can even implement a real-time gravitational wave signals detection using the deep learning algorithms which is more effective than conventional match-filtering techniques.

The advancement of artificial intelligence creates abrupt leaps in the fields, opens entire new branches of investigation, especially in the field of gravitational wave physics. In turn, it will accelerate the development of next-generation detectors and deepen our understanding of the structure of the universe as we widen up the window in space exploration.

REFERENCES

- [1] J. B. Hartle. *Gravity: an introduction to Einstein's general relativity*. Addison-Wesley, San Francisco, 2003.
- [2] Dingyun Zhang. Detecting Gravitational Waves using Constant-Q Transform and Convolutional Neural Networks. In *2021 The 4th International Conference on Computational Intelligence and Intelligent Systems*, pages 37–43, Tokyo Japan, November 2021. ACM.
- [3] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Technical Report arXiv:1905.11946, arXiv, September 2020. arXiv:1905.11946 [cs, stat] type: article.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. Number: arXiv:1512.03385 arXiv:1512.03385 [cs].
- [5] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. Technical Report arXiv:1608.06993, arXiv, January 2018. arXiv:1608.06993 [cs] type: article.
- [6] Bradley W. Carroll and Dale A. Ostlie. *An introduction to modern astrophysics*. Pearson, Harlow, 2. ed., pearson new internat. ed edition, 2014.
- [7] Andrzej Królak and Mandar Patil. The first detection of gravitational waves. *Universe*, 3(3), 2017.
- [8] Michelson-Morley experiment | Description, Results, & Facts | Britannica.
- [9] LIGO Hanford.
- [10] What do you mean by Convolutional Neural Network? - Data Science, AI and ML, June 2021.
- [11] Abien Fred Agarap. Deep Learning using Rectified Linear Units (ReLU), February 2019. Number: arXiv:1803.08375 arXiv:1803.08375 [cs, stat].
- [12] Taweh Beysolow II. *Introduction to Deep Learning Using R*. Apress, Berkeley, CA, 2017.
- [13] Plamen G. Krastev. Real-time detection of gravitational waves from binary neutron stars using artificial neural networks. *Physics Letters B*, 803:135330, April

2020.

- [14] Wei Wei and E.A. Huerta. Deep learning for gravitational wave forecasting of neutron star mergers. *Physics Letters B*, 816:136185, May 2021.
- [15] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, 45:516–535, September 2017.
- [16] Joydwip Mohajon. Confusion Matrix for Your Multi-Class Machine Learning Model, July 2021.
- [17] Abraham Pais. *"Subtle is the Lord—": the science and the life of Albert Einstein*. Oxford University Press, Oxford ; New York, 2005.
- [18] Brian Clegg. *Gravitational waves: how Einstein's spacetime ripples reveal the secrets of the universe*. Hot science. Icon Books Ltd, London, 2018.
- [19] Mike Guidry. *Modern general relativity: black holes, gravitational waves, and cosmology*. Cambridge University Press, Cambridge New York, NY Melbourne New Delhi Singapore, 2019.
- [20] Jorge Cervantes-Cota, Salvador Galindo-Uribarri, and George Smoot. A Brief History of Gravitational Waves. *Universe*, 2(3):22, September 2016.
- [21] O Gron and S Hervik. Einstein's General Theory of Relativity. page 529.
- [22] Zhao Wen, Zhang Xing, Liu Xiao-jin, Zhang Yang, Wang Yun-yong, Zhang Fan, Zhao Yu-hang, Guo Yue-fan, Chen Yi-kang, Ai Shun-ke, Zhu Zong-hong, Wang Xiao-ge, Lebigot Eric, Du Zhi-hui, Cao Jun-wei, Qian Jin, Yin Cong, Wang Jian-bo, Blair David, Ju Li, Zhao Chun-nong, and Wen Lin-qing. Gravitational Waves and Gravitational-wave Sourcedtwo. *Chinese Astronomy and Astrophysics*, 42(4):487–526, October 2018.
- [23] Bernard F. Schutz. *A first course in general relativity*. Cambridge University Press, Cambridge ; New York, 2nd ed edition, 2009.
- [24] B.P. Abbott, R. Abbott, T.D. Abbott, M.R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R.X. Adhikari, V.B. Adya, C. Affeldt, M. Agathos, K. Agatsuma, N. Aggarwal, O.D. Aguiar, L. Aiello, ... LIGO Scientific Collaboration Ain, and Virgo Collaboration. Observation of Gravitational Waves from a Binary Black Hole Merger. *Physical Review Letters*, 116(6):061102, February 2016.
- [25] LIGO's First-Ever Detection of Gravitational Waves Opens a New Window on the Universe | WIRED.
- [26] Sarah Kramer. This collision was 50 times more powerful than all the stars in the universe combined, February 2016.
- [27] Davide Castelvecchi and Alexandra Witze. Einstein's gravitational waves found

at last. *Nature*, page nature.2016.19361, February 2016.

- [28] Elena Cuoco, Jade Powell, Marco Cavaglià, Kendall Ackley, Michal Bejger, Chayan Chatterjee, Michael Coughlin, Scott Coughlin, Paul Easter, Reed Essick, Hunter Gabbard, Timothy Gebhard, Shaon Ghosh, Leila Haegel, Alberto Iess, David Keitel, Zsuzsa Marka, Szabolcs Marka, Filip Morawski, Tri Nguyen, Rich Ormiston, Michael Puerrer, Massimiliano Razzano, Kai Staats, Gabriele Vajente, and Daniel Williams. Enhancing Gravitational-Wave Science with Machine Learning. *Machine Learning: Science and Technology*, 2(1):011002, December 2020. arXiv:2005.03745 [astro-ph, physics:gr-qc].
- [29] Heming Xia, Lijing Shao, Junjie Zhao, and Zhoujian Cao. Improved deep learning techniques in gravitational-wave data analysis. *Physical Review D*, 103(2):024040, January 2021. arXiv:2011.04418 [astro-ph, physics:gr-qc, stat].
- [30] Hunter Gabbard, Michael Williams, Fergus Hayes, and Chris Messenger. Matching matched filtering with deep networks in gravitational-wave astronomy. *Physical Review Letters*, 120(14):141103, April 2018. arXiv:1712.06041 [astro-ph, physics:gr-qc].
- [31] Kaggle.com. G2Net Gravitational Wave Detection.
- [32] Nola Taylor Tillman and Ailsa Harvey published. What Is Cosmology? Definition & History, February 2022.
- [33] Peter Coles. *Cosmology: a very short introduction*. Number 51 in Very short introductions. Oxford University Press, Oxford ; New York, 2001. OCLC: ocm46984946.
- [34] Herbert Goldstein. *Classical mechanics*. Addison-Wesley series in physics. Addison-Wesley Pub. Co, Reading, Mass, 2d ed edition, 1980.
- [35] Cornelius Lanczos. *The variational principles of mechanics*. Dover books on physics and chemistry. Dover Publications, New York, 4th ed edition, 1986.
- [36] Stephen T. Thornton and Andrew F. Rex. *Modern physics for scientists and engineers*. Cengage Learning, Boston, MA, 4th ed edition, 2013.
- [37] Russell Stannard. *Relativity: a very short introduction*. Number 190 in Very short introductions. Oxford University Press, Oxford ; New York, 2008. OCLC: ocn216938492.
- [38] Andrew J S Hamilton. General Relativity, Black Holes, and Cosmology. page 1108.
- [39] J. Craig Wheeler. *Cosmic catastrophes: exploding stars, black holes, and mapping the universe*. Cambridge University Press, Cambridge ; New York, 2nd ed edition, 2007. OCLC: ocm73954922.
- [40] Jojo Moolayil. *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*. Apress, Berkeley, CA, 2019.

- [41] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation Functions: Comparison of trends in Practice and Research for Deep Learning, November 2018. Number: arXiv:1811.03378 arXiv:1811.03378 [cs].
- [42] Bernard Mont-Reynaud. The bounded-q approach to time-varying spectral analysis. 1986.
- [43] Christian Schörkhuber and Anssi Klapuri. Constant-Q Transform Toolbox For Music Processing. page 8.
- [44] Music 101: What Is an Octave? - 2022.
- [45] nnAudio.Spectrogram.CQT1992v2 — 0.2.0.
- [46] Kafeng Wang, Xitong Gao, Yiren Zhao, Xingjian Li, Dejing Dou, and Cheng-Zhong Xu. Pay Attention to Features, Transfer Learn Faster CNNs. page 14, 2020.
- [47] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.
- [48] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, December 2016.
- [49] Ajay Arasanipalai. Weights & Biases, September 2020.
- [50] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. page 8.
- [51] Accuracy and Loss: Things to Know about The Top 1 and Top 5 Accuracy | by Anh T. Dang | Towards Data Science.
- [52] Karar Ali, Zaffar Ahmed Shaikh, Abdullah Ayub Khan, and Asif Ali Laghari. Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer. *Neuroscience Informatics*, 2(4):100034, December 2022.
- [53] Phil Kim. *MATLAB Deep Learning*. Apress, Berkeley, CA, 2017.
- [54] Hossin M and Sulaiman M.N. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):01–11, March 2015.
- [55] Kai Ming Ting. *Confusion Matrix*, pages 260–260. Springer US, Boston, MA, 2017.